Feedback-prop: Convolutional Neural Network Inference under Partial Evidence

Tianlu Wang[1], Kota Yamaguchi[2], Vicente Ordonez[1]
[1]University of Virginia, [2]CyberAgent, Inc.
yamaguchi_kota@cyberagent.co.jp
{tw8cb, vicente}@virginia.edu

Abstract

We propose an inference procedure for deep convolutional neural networks (CNNs) when partial evidence is available. Our method consists of a general feedback-based propagation approach (feedback-prop) that boosts the prediction accuracy for an arbitrary set of unknown *target labels when the values for a non-overlapping arbitrary set of target labels are* known. *We show that existing models trained in a multi-label or multi-task setting can readily take advantage of feedback-prop without any retraining or fine-tuning. Our feedback-prop inference procedure is general, simple, reliable, and works on different challenging visual recognition tasks. We present two variants of feedback-prop based on layer-wise and residual iterative updates. We experiment using several multi-task models and show that feedback-prop is effective in all of them. Our results unveil a previously unreported but interesting dynamic property of deep CNNs. We also present an associated technical approach that takes advantage of this property for inference under partial evidence in general visual recognition tasks.*

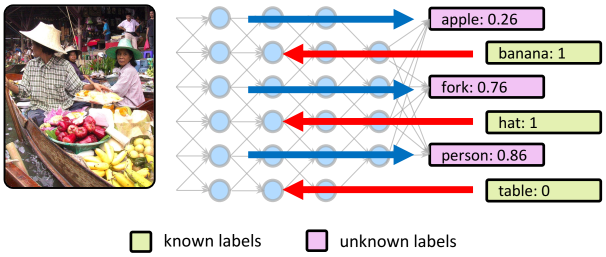


Figure 1: Feedback-prop inference leverages an arbitrary set of *known* labels to iteratively predict a set of *unknown* labels for a test input image. This example shows a multi-label classification task. Neural activations are used to transfer information among variables in the target space.

1. Introduction

In this paper we tackle visual recognition problems where partial evidence or partial information about an input image is available at test time. For instance, if we know for certain that an image was taken at the *beach*, this should change our beliefs about the types of objects that could be present, e.g. an *office chair* would be unlikely. This is because something is *known* for certain about the image even before performing any visual recognition. We argue that this setting is realistic in many applications. For instance, images on the web are usually surrounded by text, images on social media have user comments, many images contain geo-location information, images taken with portable devices contain other sensor information. More generally, images in standard computer vision datasets are effectively partially annotated with respect to a single task or modality. Assuming only visual content as inputs, while convenient for benchmarking purposes, does not reflect many end-user applications where extra information is available during inference. We propose here a general framework to address this problem in any task involving deep convolutional neural networks trained with multiple target outputs (i.e. multi-label classification) or multiple tasks (i.e. multi-task learning). We provide an example in Figure 1, where a set of labels are *known*: `banana`, `hat`, `table`, while we are trying to predict the other labels: `apple`, `fork`, `person`.

Convolutional neural networks (CNNs) have become the state-of-the-art in most visual recognition tasks. Their extraordinary representation ability has allowed researchers to address problems at an unprecedented scale with remarkable accuracy. While reasoning under partial evidence using probabilistic graphical models would involve marginalization over the variables of interest, CNNs do not model a joint distribution, therefore making such type of reasoning non-trivial. The typical pipeline using CNNs for visual recognition involves training the model using stochastic gradient descent (SGD) and the back-propagation algorithm [30] using an annotated image dataset, and then performing forward-propagation during inference given only visual input. In this paper, we challenge this prevailing inference procedure in CNNs where information only flows in one direction, and the model structure is static and fixed after training. We propose instead feedback-based propagation (feedback-prop) where forward and backward-propagation steps use intermediate neural activations to share information among output variables during inference. We show the effectiveness of our approach on multi-label prediction under incomplete and noisy labels, hierarchical scene categorization, and multi-task learning with object annotations and image descriptions.

Our main hypothesis is that by *correcting* an intermediate set of neural activations using partial labels for a given input sample, we would also be able to make more accurate predictions for the complement set of *unknown* labels. We demonstrate this behavior using our feedback-prop inference for multiple tasks and under multiple CNN models. There is remarkable evidence in previous research aimed at interpreting intermediate representations in CNNs showing that they encode basic patterns of increasing visual complexity (i.e. edges, attributes, object parts, objects) that are shared among target outputs [34, 43, 10, 38, 3]. Since the underlying shared representations of a CNN capture common patterns among target outputs, we find that they can act as pivoting variables to transfer knowledge among variables in the target space. We show that feedback-prop is general, simple to implement, and can be readily applied to a variety of problems where a model is trained to predict multiple labels or multiple tasks. Our code and data are available[1].

Our contributions can be summarized as follows:

- A general feedback-based propagation inference procedure (feedback-prop) for CNN inference under partial evidence.
- Two variants of feedback-prop using layer-wise feedback updates, and residual feedback updates, and experiments showing their effectiveness on both multi-label and multi-task settings, including an experiment using in-the-wild web data.
- An extensive analysis of CNN architectures regarding optimal layers in terms of information sharing with respect to target variables using feedback-prop.

2. Related Work

Use of Context in Computer Vision Using contextual cues in visual recognition tasks has long been studied in the psychology literature [26, 25, 4, 7, 2], and some of these insights have also been used in computer vision [28, 12, 9, 23, 18]. However, unlike our paper, most previous works using context still assume no extra information about images during inference. Instead, contextual information is predicted jointly with target variables, and is often used to impose structure in the target space based on learned priors, label relation ontology, or statistics. In contrast, our work leverages during inference the underlying contextual relations that are already implicitly learned by a CNN.

Conditional Inference in Graphical Models Our work also has connections with graphical models where messages are iteratively passed through nodes in a learned model that represents a joint distribution [24, 31]. In our inference method, messages are passed between nodes in a convolutional neural network in forward and backward directions using gradients, intermediate activations, as well as additional residual variables.

Multi-task Learning Another form of using context is by jointly training on multiple correlated visual recognition tasks or multi-task learning [29, 39, 20], where knowledge about one task helps another target task. Our inference method is highly complementary and especially useful with these types of models as it can directly be used when extra information is available for at least one of the tasks or modalities. Unlike simple conditional models that would require re-training under a fixed set of conditional input variables, feedback-prop may be used with an arbitrary set of target variables, and does not require re-training.

Optimizing the Input Space In terms of technical approach, feedback-prop has connections to previous works that optimize over inputs. One prominent example is the generation of adversarial examples that are constructed to fool a CNN model [15]. This style of gradient-based optimization over inputs is also leveraged in the task of image style transfer [13]. Gradients over inputs are also used as the supervisory signal in the generator network of Generative Adversarial Networks (GANs) [14]. Gradient-based optimization has also been used to visualize, identify, or interpret the intermediate representations learned by a deep CNN [34, 6, 42, 44, 32, 5]. However, unlike these methods, we are still interested in the target predictions and not the inputs. We find that CNN layers that lie somewhere in the middle are more beneficial to optimize as pivot variables under our model than the input image.

Deep Inference under Partial Annotations In terms of setup, a relevant recent experiment was reported in Hu et al [17]. This work introduces a novel deep Structured Inference Neural Network (SINN) model that can be adapted to a setting where true values for a set of labels are *known* at test time. We compare feedback-prop against a re-implementation of SINN for fine-grained scene categorization when a set of coarse scene categories are used as *known* labels, demonstrating superior performance without additional parameters. Tag completion is another relevant problem [40], but our approach is not specific to multi-label inference and can be easily applied to multiple diverse tasks.

[1] https://github.com/uvavision/feedbackprop

3. Method

This section presents our feedback-based inference procedure. We start from the derivation of a basic single-layer *feedback-prop* inference (Sec 3.1), and introduce our two more general versions: *layer-wise feedback-prop* (LF) (Sec 3.2), and our more efficient *residual feedback-prop* (RF) (Sec 3.3).

3.1. Feedback-prop

Let us consider a feed-forward CNN already trained to predict multiple outputs for either a single task or multiple tasks. Let $\hat{Y} = F(X, \Theta)$ represent this trained CNN, where X is an input image, $\hat{Y}$ is a set of predicted output variables, and Θ are the model parameters. Now, let us assume that the true values for some output variables are *known* at inference time, and split the variables into *known* and *unknown*: $Y = (Y_{\mathrm{k}}, Y_{\mathrm{u}})$. The neural network by default makes a joint prediction for both sets of variables: $\hat{Y} = (\hat{Y}_{\mathrm{k}}, \hat{Y}_{\mathrm{u}}) = (F_{\mathrm{k}}(X, \Theta), F_{\mathrm{u}}(X, \Theta))$. Given a *known* set of true values Y_{k}, we can compute a partial loss only with respect to this set for input sample X as $L(Y_{\mathrm{k}}, \hat{Y}_{\mathrm{k}})$. The key idea behind feedback-prop is to back-propagate this partially observed loss to the network, and iteratively update the input X in order to re-compute the predictions on the set of *unknown* variables Y_{u}. Formally, our basic feedback-based procedure can be described as follows:

$$X^* = \mathrm{argmin}_X L(Y_{\mathrm{k}}, F_{\mathrm{k}}(X, \Theta)), \tag{1}$$

$$\hat{Y}_{\mathrm{u}}^* = F_{\mathrm{u}}(X^*, \Theta), \tag{2}$$

where we optimize X, which acts as our pivoting variable, and forward-propagate to compute refined *unknown* variables $\hat{Y}_{\mathrm{u}}^*$. In fact, we need not be restricted to optimize X and can generalize the formulation to optimize arbitrary intermediate representations. Let us denote the l-th layer internal neural activations of the network as a_l, and the dissected network at layer l by $Y = F^{(l)}(a_l)$, which can be interpreted as a truncated forward propagation in the original network from layer l until the output. Then, we can define *single-layer feedback-prop* as follows:

$$a_l^* = \mathrm{argmin}_{a_l} L(Y_{\mathrm{k}}, F_{\mathrm{k}}^{(l)}(a_l, \Theta)), \tag{3}$$

$$\hat{Y}_{\mathrm{u}} = F_{\mathrm{u}}^{(l)}(a_l^*, \Theta). \tag{4}$$

In this formulation, we optimize intermediate representations at an arbitrary layer in the original model shared by F_{k} and F_{u}. These intermediate neural activations act as pivoting variables. Note that equation 1 is a special case of single-layer feedback-prop when $a_0 \equiv X$.

In our description of feedback-prop we define the output space Y as a set of variables. Each output variable can be arbitrarily complex, diverse and seemingly unrelated, as is often the case in multi-task models. In the simpler scenario of multi-label prediction, each variable corresponds to a label. We illustrate in Figure 2 an overview of our feedback-prop approach for a multi-label prediction model.

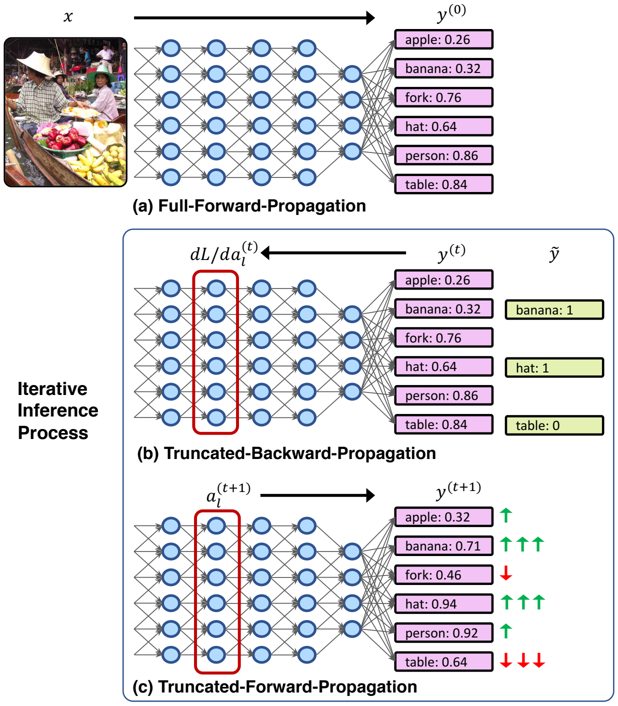


Figure 2: Overview of our feedback-prop iterative inference procedure consisting of three basic steps - (a) full forward propagation to predict initial scores for all labels, (b) truncated backward propagation to update intermediate activations based on the partial evidence (*known* labels), and (c) truncated forward propagation to update the scores for the *unknown* labels.

3.2. Layer-wise Feedback-prop (LF)

In this section we propose a more general version of feedback-prop that leverages multiple intermediate representations in a CNN across several layers: Layer-wise feedback-prop. This procedure minimizes a loss function $L(Y_{\mathrm{k}}, F_{\mathrm{k}}(A, \Theta))$ by optimizing a set of topologically sorted intermediate activation $A \equiv \{a_i, a_{i+1}, \cdots, a_N\}$ starting at layer i. However, in feed-forward models, a_l is needed to compute a_{l+1}. This requires optimizing these multiple intermediate representations using layer-by-layer sequential updates. We describe *layer-wise feedback-prop* in detail in Algorithm 1. Forward represents a truncated forward propagation from the given input at a certain layer until the output layer, and Backward represents a truncated back-propagation of gradients from the output layer to the intermediate pivoting activations. Given an input image X, known values for variables Y_{k}, and a topologically sorted

Algorithm 1 Layer-wise Feedback-prop Inference

```
Input: Input image $X$, *known* labels $Y_\mathrm{k}$, and a list of layers
    $\mathcal{L} \equiv \{i, i+1, \cdots, N\}$
Output: Prediction $\hat{Y}_\mathrm{u}$
1:  $a_0^{(T)} := X$
2:  for $l \in \mathcal{L}$ do
3:      $\hat{Y}_\mathrm{k}^{(0)}, a_l^{(0)} :=$ Forward$(a_{l-1}^{(T)})$
4:      for $t = 0$ to $T$ do
5:          Compute the partial loss $L(Y_\mathrm{k}, \hat{Y}_\mathrm{k}^{(t)})$
6:          $\frac{\partial L}{\partial a_l^{(t)}} :=$ Backward$(L)$
7:          $a_l^{(t+1)} := a_l^{(t)} - \lambda \frac{\partial L}{\partial a_l^{(t)}}$
8:          $\hat{Y}_\mathrm{k}^{(t+1)} :=$ Forward$(a_l^{(t+1)})$
9:      end for
10: end for
11: $\hat{Y}_\mathrm{u} =$ Forward$(a_N^{(T)})$
```

Algorithm 2 Residual Feedback-prop Inference

```
Input: Input image $X$, *known* labels $Y_\mathrm{k}$, and a list of layers
    $\mathcal{L} \equiv \{i, i+1, \cdots, N\}$
Output: Prediction $\hat{Y}_\mathrm{u}$
1:  $\mathbf{r}^{(0)} \equiv \{r_l^{(0)} | l \in \mathcal{L}\} := \mathbf{0}$
2:  $a_0 := X$
3:  for $t = 0$ to $T$ do
4:      for $l \in \mathcal{L}$ do
5:          $a_l^{(t)} :=$ Forward$(a_{l-1}^{(t)}) + r_l^{(t)}$
6:      end for
7:      $\hat{Y}_\mathrm{k}^{(t)} :=$ Forward$(a_N^{(t)})$
8:      Compute the partial loss $L(Y_\mathrm{k}, \hat{Y}_\mathrm{k}^{(t)})$
9:      $\frac{\partial L}{\partial \mathbf{r}^{(t)}} :=$ Backward$(L)$
10:     $\mathbf{r}^{(t+1)} := \mathbf{r}^{(t)} - \lambda \frac{\partial L}{\partial \mathbf{r}^{(t)}}$
11: end for
12: $\hat{Y}_\mathrm{u} =$ Forward$(a_N^{(T)})$
```

list of layers $\mathcal{L}$, the algorithm optimizes internal representations a_l in topological order. More generally, these layers do not need to be consecutive. The updates are performed in this fashion so that the algorithm *freezes* activation variable a_l layer-by-layer from the input side, so that after each freeze, the next variable can be initialized to apply feedback updates. In Algorithm 1, λ is an update rate and iterative SGD steps are repeated T times. The update operation (line 7) may be replaced by other types of SGD update rules such as SGD with momentum, AdaGrad, or Adam. Note that the backward, and forward propagation steps only go back as far as a_l, and do not require a full computation through the entire network. The *single-layer feedback-prop* inference in Sec 3.1 is a special case of *layer-wise feedback-prop* when $|\mathcal{L}| = 1$. The choice of layers will affect the quality of feedback-prop predictions for *unknown* targets.

3.3. Residual Feedback-prop (RF)

The proposed *layer-wise feedback-prop* (LF) inference can use an arbitrary set of intermediate layer activations, but is inefficient due to the double-loop in Algorithm 1, where layers have to be updated individually in each pass. Here, we refine our formulation even further by updating multiple layer activations in a single pass through the incorporation of auxiliary residual variables. We name this version of our inference procedure *residual feedback-prop* (RF) inference.

The core idea in RF is to inject an additive variable (feedback residual) to intermediate representation variables, and optimize over residuals instead of directly updating intermediate representations. Notice that incorporation of these residual variables takes place only during inference, and does not involve any modifications in learning, or whether the underlying model itself uses residuals. We add a feedback residual variable r_l to the unit activation a_l in the forward propagation at layer l as follows:

$$a_l = f_l(a_{l-1}, \theta_l) + r_l, \tag{5}$$

where f_l is the layer transformation function at l (e.g. convolutional filtering) with model parameters θ_l. When $r_l = 0$, this is a regular forward-propagation. Instead of directly updating a_l by feedback-prop as in LF, we only update residual variables r_l. Figure 3 shows how residual variables are incorporated in a model during inference.

Algorithm 2 describes in detail how residual feedback-prop operates. The procedure starts by setting residuals to zero (line 1). The inner-loop is a truncated feed-forward propagation starting in activation a_l but using additive residuals. Notice that this computation does not incur significant computational overhead compared to regular forward propagation. Updates do not require a double-loop (lines 9-10), therefore avoiding repetitive gradient computations as in LF. We show in our experiments that residual-based feedback-prop performs comparably to layer-wise

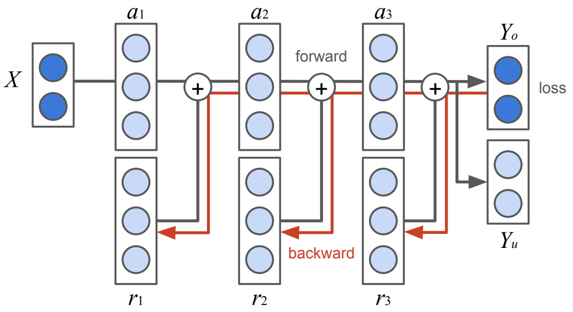


Figure 3: In our RF approach, residual variables r_l are updated instead of intermediate activations a_l in order to update all layers in a single pass.

feedback-prop in multi-label and multi-task models, and is more efficient when updating multiple layers (Sec 6).

4. Experiments

We evaluate our approach on four tasks 1) Multi-label image annotation with incomplete labels, where incomplete labels are simulated at test time by artificially splitting the total vocabulary of labels into *known* and *unknown* (Sec 4.1), 2) Hierarchical scene categorization, where true values for coarse scene categories are known and the aim is to predict fine-grained scene categories (Sec 4.2), 3) Automatic annotation of news images in-the-wild, where surrounding news text is *known*, and a set of visual words from image captions are the *unknown* targets (Sec 4.3), and 4) A multi-task joint prediction of image captions and object categories, where the goal during inference is to predict image captions as the *unknown* target (Sec 4.4).

4.1. Multi-label Image Annotation

This experiment uses the COCO dataset [22], containing around 120k images, each with 5 human-annotated captions. We use the standard split in the dataset that has $82,783$ images in the training set and subdivide the standard validation set into $20,000$ images for validation and $20,504$ for testing. Our task is to predict visual concepts for any given image similar to the visual concept classifier used by Fang et al [11], which we use as our baseline. We build a vocabulary of concepts using the most frequent 1000 words in captions from the training set after tokenization, lemmatization, and stop-word removal. We first train a multi-label prediction model by modifying a standard CNN to generate a 1000-dimensional output, and learn logistic regressors using the following loss function:

$$L = -\sum_{i=1}^{d}\frac{1}{N}\sum_{j=1}^{N}\lambda_j[y_{ij}\log\sigma(f_j(I_i,\Theta)) + (1-y_{ij})\log(1-\sigma(f_j(I_i,\Theta)))], \tag{6}$$

where $\sigma(x) = 1/(1+exp(-x))$ is the sigmoid function, $f_j(I_i,\Theta)$ is the unnormalized output score for category j given image I_i, and Θ are the model parameters of the underlying CNN. Intuitively, each term in this loss function encourages activation f_j to increase if label $y_{ij} = 1$ or decrease otherwise. Weight parameters λ_j count the contribution of each class j differently. These parameters are designed to handle the extreme class imbalance in multi-label image annotation - larger values of lambda are assigned to classes that occur less frequently. Particularly, we set $\lambda_j = \sum_{i=1}^{|D|}(1-y_{ij})\,/\sum_{i=1}^{|D|}y_{ij}$. We load weights from models pretrained on ImageNet to train our models.

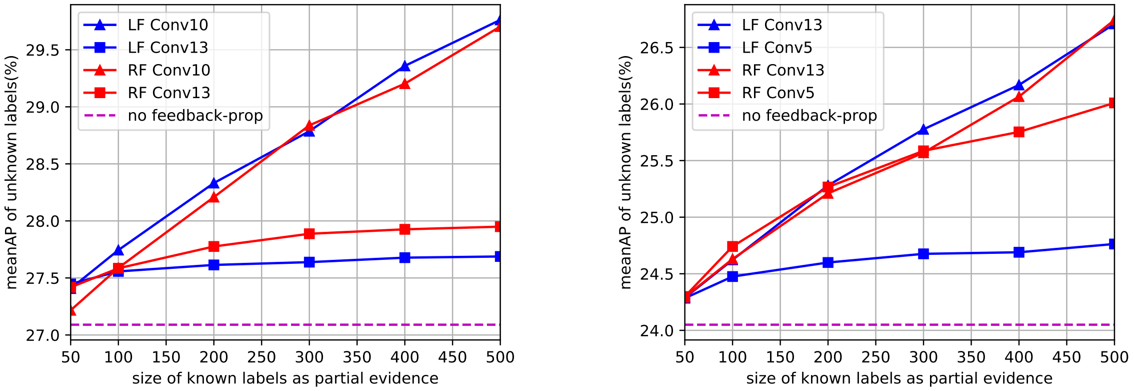
For feedback-prop evaluation, we put aside a fixed set of 500 targets as *unknown*. We measure the mean average precision, mAP, (area under the precision-recall curve) averaged on the *unknown* label set as we experiment with different amounts of *known* labels, from 50 to the total complement set of 500 labels. Figure 4 reports the results for both LF and RF, using several intermediate representations from VGG-16 [35] and Resnet-18 [16]. We determine the update rate parameter and number of iterations using the validation split, and report results on the test split. When the amount of *known* labels is less than 500, we run 5 rounds with randomly sampled labels and report average performance.

Observations: Remarkably, for both LF and RF, accuracy increases with the amount of partial evidence without any apparent diminishing returns. Different layers achieve different levels of accuracy, indicating that information shared with the target label space changes across internal convolutional layers in both Resnet-18 and VGG-16. Figure 4(a) shows that VGG-16 achieves a mAP on the set of *unknown* labels of 27.09 when using only the image as input, and the mAP is improved to 27.41 on average when only using a random sample of 50 *known* labels when using the outputs of Conv13 as pivoting variables under LF. Note that these 50 *known* labels are potentially unrelated to the 500 labels the model is trying to predict, and most of them only provide weak negative evidence (e.g. $y_{ij} = 0$). When using the full complement set of 500 labels, the predictions achieve 29.76 mAP, which represents a 9.8% relative improvement. Figure 4(b) shows that Resnet-18 achieves a mAP of 24.05 using no additional evidence. RF under Conv13 outputs as pivoting variables can reach 26.74 mAP given the non-overlapping set of 500 *known* labels as partial evidence, a relative improvement of 11.2%.

4.2. Hierarchical Scene Categorization

We apply feedback-prop on scene categorization on the SUN dataset [41]. This dataset has images annotated with 397 fine-grained scene categories, 16 general scene categories, and 3 coarse categories. We follow the same experimental setting of train, validation and test split ratio reported in [1] with 50, 10 and 40 images from every scene category. Our task is to infer fine-grained categories given true values for coarse categories as it was performed in Hu et al [17]. For evaluation, we compute multi-class accuracy (*MC Acc*) and intersection-over-union accuracy (*IoU Acc*) as well as mean average precision (mAP) averaged over all categories.

Observations: Table 1 reports results averaged over 5 runs. We use a CNN + Softmax classifier as our first Baseline, and as a second baseline a CNN + Softmax classifier that uses true values for coarse categories in the form of a binary indicator vector as additional input to the classifier (Baseline + PL). Similar baselines were used in Hu et al [17]. Additionally, we re-implement the Structured Inference Neural Network (SINN) of Hu et al [17] which outputs three lev-



(a) Feedback-prop on VGG16

(b) Feedback-prop on ResNet18

Figure 4: Performance (mAP) of LF and RF using different intermediate activations (Conv5, 10, 13) against the amount of *known* labels in the COCO multi-label image annotation task: the more the labels, the higher the performance.

	MC Acc	*mAP*	*IoU Acc*
Baseline [17]	52.83±0.24	56.17±0.21	35.90±0.22
Bsln + PL [17]	53.15±0.27	56.49±0.24	36.20±0.26
SINN + PL [17]	54.30±0.35	58.45±0.31	37.28±0.34
Ours (LF)	54.93±0.42	58.52±0.34	37.86±0.39
Ours (RF)	**55.01±0.35**	**58.70±0.26**	**37.95±0.33**

Table 1: Feedback-prop on hierarchical scene categorization in SUN397. Our methods (LF / RF) outperform baseline methods on all metrics when partial labels are available.

els of predictions for fine-grained, general, and coarse scene categories and connects them using a series of linear layers modeling positive and negative relations in the target space and in both top-down and bottom-up directions. Instead of using WordNet to estimate label relations, we threshold pearson correlation coefficients between target variables in the training split. Both LF and RF successfully outperform the baselines and the previously proposed model in all metrics. Notice that our proposed method does not require a significant amount of additional parameters. In these experiment RF and LF use as pivoting variables the outputs of Conv-{2, 3, 4, 5}. For this experiment, all models rely on Alexnet [21] pretrained in the Places365 dataset [45].

4.3. Visual Concept Prediction on News Images

In this experiment, we train a multi-task model that jointly predicts a set of visual concepts from news image captions and a separate set of concepts from surrounding text. We first collected a dataset of news images with associated captions and text from the BBC news website. Our splits have $153,364$ images for training, $10,213$ images for validation, and $10,307$ images for testing. Both tasks are trained under the same multi-label loss and setup from Sec 4.1. The vocabulary for visual concepts from image captions consists of the 500 most frequent nouns, and the vocabulary for visual concepts from surrounding news texts consists of the top $1,000$ most frequent nouns. We use Resnet-50 [16] trained under the sum of the losses for each task. At inference time, we predict the visual concepts defined by words in captions (*unknown* labels), given the input image and the surrounding news text (*known* labels). We evaluate LF using layer Conv40 and RF under Conv22 as pivoting variables respectively, which we generally find to perform best in previous experiments. Table 2 shows the mAP across the set of *unknown* labels in the test split with varying amounts of additional partial evidence (surrounding news text).

	LF-conv-40	RF-conv-22
no-text	19.92	19.92
25% text	21.33	21.27
50% text	22.16	22.23
75% text	22.42	22.51
100% text	**22.57**	**22.57**

Table 2: mAP of visual concept predictions on news images without vs with surrounding news text.

Observations: The mAP for predicting the set of *unknown* labels improves from 19.921% (only using input images) to 21.329% even when only using the first 25% of the surrounding news text as additional evidence. Using a larger portion of surrounding news text consistently increases the accuracy. When using all the available surrounding text for each news image the mAP improves on average from 19.92% to 22.57%, a relative improvement of 13.3%. This is remarkable since –unlike our previous experiment– the surrounding text might also contain many confounding signals and noisy labels. We show qualitative examples of LF using all surrounding text as partial evidence in Figure 6.

	LF	RF
no-fp	26.98	26.98
fp-input	29.14	29.53
fp-conv-1	29.72	29.56
fp-conv-4	29.65	29.66
fp-conv-7	29.77	**29.79**
fp-conv-10	**29.82**	29.74
fp-conv-13	27.59	27.87

Table 3: VGG-16 layer-wise analysis.

	LF	RF
no-fp	24.08	24.08
fp-input	24.74	27.06
fp-conv-1	24.16	25.91
fp-conv-5	24.57	25.76
fp-conv-9	25.94	26.71
fp-conv-13	**26.80**	**27.26**
fp-conv-17	24.19	24.22

Table 4: Resnet-18 layer-wise analysis.

	LF	RF
no-fp	26.94	26.94
fp-input	28.35	29.28
fp-conv-1	27.60	29.49
fp-conv-10	29.54	29.80
fp-conv-22	29.61	**29.89**
fp-conv-40	**29.71**	29.67
fp-conv-49	27.14	27.14

Table 5: Resnet-50 layer-wise analysis.

4.4. Joint Captioning and Object Categorization

We train a multi-task CNN model on the COCO dataset [22] to jointly perform caption generation and multi-label object categorization. We use Resnet-50 with two additional output layers after the last convolutional layer: a multi-label prediction layer with 80-categorical outputs corresponding to object annotations, and an LSTM decoder for caption generation as proposed by Vinyals et al [37]. We shuffle images in the standard COCO train and validation splits and use 5000 images for validation and test, and the remaining samples for training. We perform the same preprocessing on images and captions as in [19]. We report BLEU[27], METEOR[8] and CIDEr[36] scores for captioning and mean average precision(mAP) for object categorization. This model achieves a 0.939 CIDEr score and 71.3% *mAP*. In order to evaluate feedback-prop, we use object annotations as *known* and analyze the effects on the quality of the predicted captions – our *unknown* target. Table 6 presents results under this regime on the test split.

	BLEU-4	ROUGE	CIDEr
no-fp [37]	28.65	0.5267	0.9466
LF-input	29.20	0.5290	0.9647
LF-conv-10	29.78	0.5333	0.9859
LF-conv-22	29.71	0.5327	0.9834
LF-conv-40	29.66	0.5332	0.9854
LF-conv-10, 40	29.73	0.5329	0.9872
RF-conv-10, 40	29.63	0.5337	0.9922

Table 6: Feedback-prop in multi-task learning: caption generation results benefit from object annotations as partial evidence using feedback-prop.

Observations: Feedback propagation between target outputs and intermediate representations (including inputs) helps generate better image captions. We observe that using LF with any layer as pivot, improves the predictions under all standard metrics. Furthermore, we observe that jointly using the outputs of layers Conv10 and Conv40 as pivots can outperform updating the outputs of any single layer. RF on Conv10 and Conv40 reaches the highest CIDEr score, improving from 0.946 to 0.992.

5. What Layers are the Most Useful?

In this section, we analyze where are the most useful intermediate representations in a CNN under feedback-prop. In other words, what are the intermediate layers of a CNN that seem to allow maximal sharing of information among target predictions. We first train three multi-label models based on Resnet-18, Resnet-50, and VGG-16 on the COCO multi-label task from Sec 4.1. For each model we report in tables 3, 4, and 5 the best validation accuracy that can be reached with the outputs of several individual layers as pivots using both LF and RF. We observe that in both VGG and Resnets, middle layers seem to be the most useful compared to layers closer to inputs or outputs. Specifically, we find that Conv13 in Resnet-18, Conv20 and Conv40 in Resnet-50, and Conv7 and Conv10 in VGG-16 achieve the best performance given the same amount of partial evidence (a fixed set of 500 *known* labels and 500 *unknown* labels). These results seem analogous to a recent study on neural networks where mutual information between intermediate representations with respect to both inputs and outputs is analyzed during training [33]. It would be interesting to devise an approach to automatically identify what layers are most effective to use as pivots under feedback-prop using an information theoretic approach.

6. Computational Efficiency

Here, we benchmark our two proposed feedback-prop methods. We use Resnet-50 multi-label model of Sec 4.1 and select a sequence of layers including *input image*,

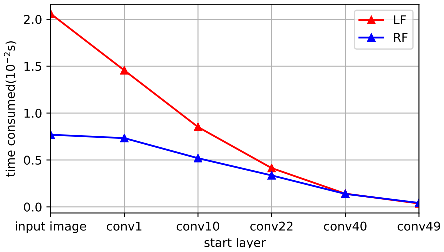


Figure 5: Benchmark results for LF and RF. The x-axis shows the earliest layer used, after which all the layers are updated. RF becomes efficient as more layers are used.

no feedback-prop predictions:

claim:0.891679	school:0.060947	try:0.319411	official:0.790290	ceremony:0.506596	people:0.494557
try:0.592581	people:0.054434	show:0.186112	home:0.310297	thousand:0.159579	light:0.325617
attack:0.278426	light:0.050388	scene:0.158961	child:0.180287	pay:0.132895	launch:0.279506
city:0.155168	part:0.045863	news:0.110425	people:0.139492	game:0.104834	sir:0.270729
hundred:0.133139	force:0.043337	people:0.092683	woman:0.088490	deal:0.080287	point:0.243272
woman:0.120313	area:0.042076	attack:0.059946	house:0.076746	people:0.071572	leave:0.150900
police:0.119733	include:0.042012	pay:0.050996	**camp:0.064999**	open:0.048961	centre:0.133657
report:0.104096	security:0.039852	lead:0.049296	use:0.063372	city:0.046278	campaign:0.110601

with feedback-prop predictions:

claim:0.913860	**clash:0.948569**	try:0.385340	**camp:0.925969**	**school:0.858543**	**vote:0.488819**
attack:0.910921	protester:0.774579	protest:0.260692	**refugee:0.908903**	game:0.284368	campaign:0.447369
bomb:0.267836	pro:0.520027	medium:0.130189	home:0.293703	play:0.234772	people:0.388327
try:0.240699	security:0.405497	china:0.119549	child:0.255574	thousand:0.112460	centre:0.309245
body:0.159527	force:0.176731	**court:0.100340**	woman:0.147657	parent:0.085781	ireland:0.271122
woman:0.123605	**police:0.159598**	show:0.086785	people:0.104480	people:0.076458	leave:0.263814
relative:0.121821	anti:0.122141	**police:0.069903**	syria:0.088542	start:0.061948	point:0.179191
militant:0.119986	government:0.064173	woman:0.067833	official:0.061292	celebrate:0.058791	**minister:0.133364**

news text labels:

people, government,	country, work, part,	people, government,	action, start, fund,	union, today, secretary,	prime, start, statement,
tell, police, country,	party, minister, report,	tell, police, country,	price, move,	offer, speak, key,	mark, station, act,
state, group, report,	number, school, leader,	part, family, child,	technology, syria,	executive, education,	person, age, return,
find, place, school,	news, meet, house,	party, group, report,	thousand, name, risk,	parent, development,	ireland, morning,
public, news, attack,	force, court, power,	company, president,	offer, hope, saw, food,	stop, radio, energy,	provide, island, couple,
force, want, official,	want, official, end,	need, leader, public,	face, education, girl,	visit, mile, everyone,	poll, candidate,
mean, support, death,	council, support,	news, business, house,	act, crime, course,	space, stage, club,	referendum, amount,
security, put, use,	election, death,	help, force, court, case,	violence, crisis, book,	opportunity, trust,	ask, voter, protect,
leave, market,	security, use, win,	member, want, official,	age, return, france,	department, sport,	date, proposal, bst,
authority, office, claim,	university, street, vote,	china, set, death,	organisation, space,	teacher, target, sir,	citizen, sex, difference,
play, town, body, air,	authority, office, fire,	security, hold, team,	access, try, hundred,	commission, football,	agree, one, limit,
agency, india, past, …	term, remain, prime, …	street, men, look, …	provide, …	position, majority, …	contract, count, …

Figure 6: Qualitative examples for visual concept prediction for News Images. Second row shows results of a multi-label prediction model (no feedback-prop), the next row shows results obtained using LF where words from surrounding news text (shown in blue) are used as partial evidence. Predictions also among the true labels are highlighted in bold. While news text contains many words that seem marginally relevant, feedback-prop still leverages them effectively to improve predictions. Surrounding news text provides high-level feedback to make predictions that would otherwise be hard.

conv1, *conv10*, *conv22*, *conv40*, and *conv49*. We pick one layer as initial layer and update this layer with all subsequent layers. For example, if *conv40* is the initial layer, we also update *conv49*. We use a single 12GB NVIDIA Pascal Titan X GPU and record average inference times per image per iteration. Figure 5 shows that as more layers are used as pivots, RF shows the more gains over LF. RF is generally faster, with a slight increase in memory footprint.

7. Conclusions

In the context of deep CNNs, we found that by optimizing the intermediate representations for a given input sample during inference with respect to a subset of the target variables, predictions for all target variables improve their accuracy. We proposed two variants of a feedback propagation inference approach to leverage this dynamic property of CNNs and showed their effectiveness for making predictions under partial evidence for general CNN models trained in a multi-label or multi-task setting. As multi-task models trained to solve a wide array of tasks such as UberNet [20] emerge, we expect a technique such as feedback-prop will become increasingly useful. An interesting future direction would be devising an approach that leverages feedback-based updates during training.

Acknowledgements This work was partially supported by a Google Faculty Research Award in Machine Perception.

References

[1] P. Agrawal, R. B. Girshick, and J. Malik. Analyzing the performance of multilayer neural networks for object recognition. In *ECCV*, 2014. 5

[2] M. Bar. Visual objects in context. *Nature Reviews Neuroscience*, 5(8):617–629, 2004. 2

[3] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *CVPR*, 2017. 2

[4] I. Biederman, R. J. Mezzanotte, and J. C. Rabinowitz. Scene perception: Detecting and judging objects undergoing relational violations. *Cognitive psychology*, 14(2):143–177, 1982. 2

[5] A. Binder, G. Montavon, S. Bach, K.-R. Müller, and W. Samek. Layer-wise Relevance Propagation for Neural Networks with Local Renormalization Layers. *arXiv:1604.00825*, 2016. 2

[6] C. Cao, X. Liu, Y. Yang, Y. Yu, J. Wang, Z. Wang, Y. Huang, L. Wang, C. Huang, W. Xu, et al. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In *ICCV*, pages 2956–2964, 2015. 2

[7] M. M. Chun and Y. Jiang. Contextual cueing: Implicit learning and memory of visual context guides spatial attention. *Cognitive psychology*, 36(1):28–71, 1998. 2

[8] M. Denkowski and A. Lavie. Meteor universal: Language specific translation evaluation for any target language. In *In Proceedings of the Ninth Workshop on Statistical Machine Translation*, 2014. 7

[9] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert. An empirical study of context in object detection. In *CVPR*, pages 1271–1278. IEEE, 2009. 2

[10] V. Escorcia, J. C. Niebles, and B. Ghanem. On the relationship between visual attributes and convolutional networks. In *CVPR*, pages 1256–1264, 2015. 2

[11] H. Fang, S. Gupta, F. N. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, C. L. Zitnick, and G. Zweig. From captions to visual concepts and back. In *CVPR*, pages 1473–1482, 2015. 5

[12] C. Galleguillos, A. Rabinovich, and S. Belongie. Object categorization using co-occurrence, location and appearance. In *CVPR*, pages 1–8. IEEE, 2008. 2

[13] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *CVPR*, pages 2414–2423, 2016. 2

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014. 2

[15] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 2

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CVPR*, 2016. 5, 6

[17] H. Hu, G.-T. Zhou, Z. Deng, Z. Liao, and G. Mori. Learning structured inference neural networks with label relations. In *CVPR*, pages 2960–2968, 2016. 2, 5, 6

[18] J. Johnson, L. Ballan, and F.-F. Li. Love thy neighbors: Image annotation by exploiting image metadata. *ICCV*, 2015. 2

[19] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 7

[20] I. Kokkinos. Ubernet: Training a 'universal' convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. *CVPR*, 2017. 2, 8

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012. 6

[22] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 5, 7

[23] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, June 2014. 2

[24] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 467–475. Morgan Kaufmann Publishers Inc., 1999. 2

[25] D. Navon. Forest before trees: The precedence of global features in visual perception. *Cognitive psychology*, 9(3):353–383, 1977. 2

[26] S. E. Palmer. The effects of contextual scenes on the identification of objects. *Memory & Cognition*, 3:519–526, 1975. 2

[27] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. 7

[28] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *ICCV*, pages 1–8. IEEE, 2007. 2

[29] W. Ruan and E. L. Miller. Ensemble multi-task gaussian process regression with multiple latent processes. *arXiv*, 2017. 2

[30] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988. 1

[31] R. Salakhutdinov and G. E. Hinton. Deep boltzmann machines. In *AISTATS*, 2009. 2

[32] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *arXiv preprint arXiv:1610.02391*, 2016. 2

[33] R. Shwartz-Ziv and N. Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017. 7

[34] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 2

[35] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015. 5

[36] R. Vedantam, C. L. Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. *CoRR*, abs/1411.5726, 2014. 7

[37] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015. 7

[38] S. Vittayakorn, T. Umeda, K. Murasaki, K. Sudo, T. Okatani, and K. Yamaguchi. Automatic attribute discovery with neural activations. In *ECCV*, pages 252–268. Springer, 2016. 2

[39] J. Wang, X. Zhu, S. Gong, and W. Li. Attribute recognition by joint recurrent learning of context and correlation. *arXiv*, 2017. 2

[40] L. Wu, R. Jin, and A. K. Jain. Tag completion for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):716–727, 2013. 2

[41] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, pages 3485–3492. IEEE, 2010. 5

[42] L. Xie, L. Zheng, J. Wang, A. L. Yuille, and Q. Tian. Interactive: Inter-layer activeness propagation. In *CVPR*, pages 270–279, 2016. 2

[43] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833. Springer, 2014. 2

[44] J. Zhang, Z. Lin, S. X. Brandt, Jonathan, and S. Sclaroff. Top-down neural attention by excitation backprop. In *ECCV*, 2016. 2

[45] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 6