

Attention-Based Prototypical Learning Towards Interpretable, Confident and Robust Deep Neural Networks

Sercan Ö. Arik¹ Tomas Pfister¹

Abstract

We propose a new framework for prototypical learning that bases decision-making on few relevant examples that we call *prototypes*. Our framework utilizes an attention mechanism that relates the encoded representations to determine the prototypes. This results in a model that: (1) enables interpretability by outputting samples most relevant to the decision-making in addition to outputting the classification results; (2) allows confidence-controlled prediction by quantifying the mismatch across prototype labels; (3) permits detection of distribution mismatch; and (4) improves robustness to label noise. We demonstrate that our model is able to maintain comparable performance to baseline models while enabling all these benefits.

1. Introduction

Deep neural networks yield excellent performance in numerous tasks, from image classification (He et al., 2016) to text classification (Conneau et al., 2016). Yet, high classification performance is not always a sufficient factor for adoption. In addition, ideally an artificial intelligence system would:

- Build trust by explaining rationales behind decisions,
- Allow detection of common failure cases and biases,
- Refrain from making decisions without sufficient confidence (and allow the required level of confidence to be adjusted according to the requirements),
- Be robust to slight changes in the training/testing setup, and against adversarial attacks.

For some application areas such as healthcare and financial services, these aspects are especially crucial. In their current form, deep neural networks are essentially considered *black-box models* – they are controlled by complex nonlinear interactions between many parameters that are difficult to understand. There are numerous approaches, e.g. (Kim

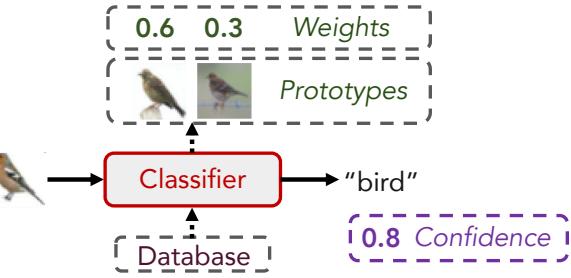


Figure 1. Prototypical learning framework for image classification. The classifier bases the classification decision on the few prototypes it chooses from the database. This enables interpretability of the prediction (by visualizing the highest weight prototypes) and confidence estimation for the decision (by measuring agreement across prototype labels).

et al., 2018; Erhan et al., 2009; Zeiler & Fergus, 2013; Simonyan et al., 2013), that enable post-hoc explainability for already-trained models. Yet, these have the fundamental limitation that the models are not designed or trained for interpretability purposes. There are also various studies on the redesign of neural networks, similar to the direction in this paper. One notable example is the sequential attention mechanism (Bahdanau et al., 2015) that restricts the focus into a small portion of the input. Another one is capsule networks (Sabour et al., 2017) that use a dynamic routing mechanism to extract the amount of detail to be processed at the next higher layer. Finally, in (Zhang et al., 2018) convolutional filters are modified to be more interpretable by encouraging low entropy for inter-category activations and spatial distributions.

In this paper we propose a new architecture and learning framework that enables interpretability of deep neural networks without deteriorating their performance. The specific form of interpretability we focus on is *prototypical learning* that sparsely decomposed the decision making (see Fig. 1). Similar to previous work (Bien & Tibshirani, 2012), we define prototypes as *the minimal subset of samples with high interpretable value that can serve as a distillation or condensed view of a dataset*. Given that the number of objects an average human can interpret is limited (Miller, 1956), outputting prototypes can be a very effective approach for

¹Google Cloud AI. Correspondence to: Sercan Ö. Arik <soarik@google.com>.

interpretability. In addition to interpretability of decision making, prototypical learning: (1) provides a very efficient confidence metric by measuring mismatches in prototype labels, allowing performance to be improved by refraining from making predictions in the absence of sufficient confidence; (2) helps detect deviations in the test distribution; and (3) enables performance in the high label noise regime to be improved by controlling the number of selected prototypes. Given these motivations, prototypes should be:

- Chosen from the samples with the accurate label,
- Perceptually relevant to the input given the task,
- Sparse (i.e. low in count),
- Robust to slight changes in the learning framework.

Prototype selection in its naive form is computationally expensive and perceptually challenging (Bien & Tibshirani, 2012). We propose a novel and efficient method for selecting input-dependent prototypes that integrates an attention mechanism between the input and prototype candidates from a database. The outputs of the attention mechanism determine prototype weights that scale the encoded representations which are consequently combined linearly. We demonstrate that the sparsity can be efficiently imposed via the choice of the attention normalization and additional regularization. We present our results for image, text and tabular data, and demonstrate that our method enables four key benefits (interpretability, confidence control, diagnosis of distribution mismatch, robustness against label noise) while maintaining comparable overall accuracy.

2. Related Work

2.1. Prototypical learning

The principles of our prototypical learning framework is based on (Bien & Tibshirani, 2012). They formulate prototype selection as an integer program and solve it using a greedy approach with linear program relaxation. It seems unclear whether such approaches can be efficiently adopted to deep neural networks. (Chen et al., 2018) and (Li et al., 2018) modify convolutional neural networks with a prototype layer for interpretability by replacing the conventional inner product with the squared L^2 distance computation for perceptual similarity. In contrast, our framework uses an attention mechanism to quantify perceptual similarity and can choose input-dependent prototypes from a large-scale candidate database. (Yeh et al., 2018) decomposes the prediction into a linear combination of activations of training points for interpretability using representer values. The linear decomposition idea also exists in our framework, but the weights are learned via an attention mechanism and sparsity is encouraged in the decomposition. In (Koh & Liang, 2017), the training points that are the most responsible for a given prediction are identified using influence functions via

oracle access to gradients and Hessian-vector products.

2.2. Metric learning

Metric learning aims to find an embedding representation of the data where similar data points are close and dissimilar data pointers are far from each other. Our framework is motivated by efficient learning of such an embedding space which can be used to decompose decisions. Metric learning for deep neural networks is typically based on modifications to the objective function, such as using triplet loss and N-pair loss (Sohn, 2016; Cui et al., 2016; Hoffer & Ailon, 2014). These yield perceptually meaningful embedding spaces yet typically require a large subset of nearest neighbors to avoid degradation in performance (Cui et al., 2016). (Kim et al., 2018) proposes a deep metric learning framework which employs an attention-based ensemble with a divergence loss so that each learner can attend to different parts of the object. Our framework has metric learning capabilities like relating similar data points, but also performs well on the ultimate supervised learning task.

2.3. Attention-based few-shot learning

Some of our inspirations are based on recent advances in attention-based few-shot learning. In (Vinyals et al., 2016), an attention mechanism is used to relate an example with candidate examples from a support set using a weighted nearest-neighbor classifier applied within an embedding space. In (Ren et al., 2018), incremental few-shot learning is implemented using an attention attractor network on the encoded and support sets. In (Snell et al., 2017), a non-linear mapping is learned to determine the prototype of a class as the mean of its support set in the embedding space. During training, the support set is randomly sampled to mimic the inference task. Overall, the attention mechanism in our framework follows related principles but fundamentally differs in that few-shot learning aims for generalization to unseen classes whereas the goal of our framework is robust and interpretable learning for seen classes.

2.4. Uncertainty and confidence estimation

Our framework takes a novel perspective on the perennial problem of quantifying how much deep neural networks' predictions can be trusted. Common approaches are based on using the scores from the prediction model, such as the probabilities from the softmax layer of a neural network, yet it has been shown that the raw confidence values are typically poorly calibrated (Guo et al., 2017). In (Papernot & McDaniel, 2018), the intermediate representations of the network are used to define a distance metric, and a confidence metric is proposed based on the conformity of the neighbors. In (Jiang et al., 2018), a confidence metric is proposed based on the agreement between the classifier and

a modified nearest-neighbor classifier on the test sample. Another direction of uncertainty and confidence estimation is Bayesian neural networks that return a distribution over the outputs (Kendall & Gal, 2017) (Mullacherry et al., 2018).

3. Proposed Learning Framework

3.1. Learning principles

Consider a training set with labels, $\{\mathbf{x}_i, y_i\}$. Conventional supervised learning aims to learn a model $s(\mathbf{x}_i; \mathbf{S})$ that minimizes a predefined loss $E\{L(y_i, \hat{y}_i = s(\mathbf{x}_i; \mathbf{S}))\}$.¹ Our goal is to impose that decision making should be based on only a few training examples, referred to as *prototypes*, such that their linear superposition in an embedding space can yield the overall decision and the superposition weights correspond to their importance factors. This framework requires decomposition of $s()$ into multiple functions of the desired form based on the following six principles:

- i. $f(\mathbf{x}_i; \theta) = \mathbf{v}_i$ should encode all relevant information of \mathbf{x}_i for the final decision. $f()$ considers the global distribution of the samples, i.e. learns from all $\{\mathbf{x}_i, y_i\}$.
- ii. The decision $\hat{y}_i = g(\mathbf{v}_i; \eta)$ should be close to the ground truth as quantified by a loss $E\{L(y_i, \hat{y}_i)\}$.
- iii. Given candidates $\mathbf{x}_j^{(c)}$ to select the prototypes from, there exists weights p_j (where $p_j \geq 0$ and $\sum_{j=1}^D p_j = 1$), such that the decision $\hat{c}_i = g(\sum_{j=1}^D p_j \mathbf{v}_j^{(c)}; \eta)$ (where $f(\mathbf{x}_j^{(c)}; \theta) = \mathbf{v}_j^{(c)}$) should be close to the ground truth as quantified by $E\{L(y_i, \hat{y}_i)\}$.
- iv. When the linear combination $\sum_{j=1}^D p_j \mathbf{v}_j^{(c)}$ is considered, prototypes with higher weights p_j should have higher contribution in the decision $g(\sum_{j=1}^D p_j \mathbf{v}_j^{(c)}; \eta)$.
- v. The weights should be sparse – only a controllable amount of weights p_j should be non-zero.
- vi. The weights p_j should depend on the relation between input and the candidate samples, $p_j = r(\mathbf{x}_i, \mathbf{x}_j^{(c)}; \Gamma)$.

Training involves optimization of the parameters θ, Γ, η .

3.2. Network architecture and training

Based on the principles above, we propose the architecture in Fig. 2. A trainable encoder is employed to transform B input samples (note that B may be 1 at inference) and D samples from the database of prototype candidates (note that D may be as large as the entire training dataset at inference) into keys, queries and values. The encoder architecture can be based on any trainable discriminative feature mapping function (e.g. ResNet (He et al., 2016) for images) with the only difference that it generates three types of embeddings. The encoder is shared and jointly updated for the input sam-

¹ \mathbf{S} represents the trainable parameters for the function $s(\cdot; \mathbf{S})$ and is sometimes not shown for notational convenience.

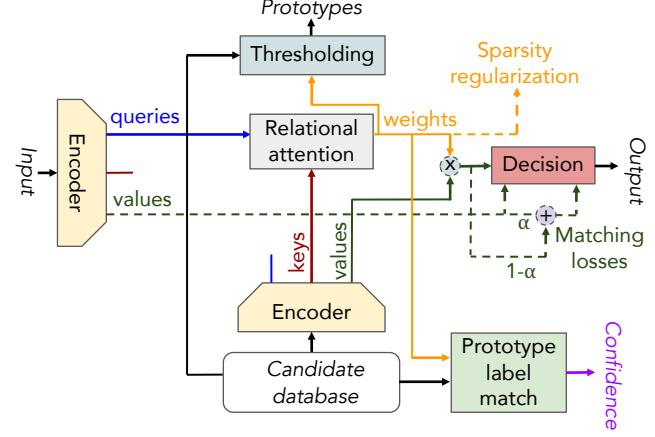


Figure 2. Prototype learning network. Dotted lines are only used while training. The encoder is shared between input samples and the candidate database samples to obtain key, query and value representations. The alignment between keys and queries determines the weights of the prototypes, and the linear combination of the values determines the final decision. Conformity of the prototype labels is used as a confidence metric.

ples and prototype candidate database. For input samples, $\mathbf{V}_b \in \mathbb{R}^{B \times d_{out}}$ and $\mathbf{Q}_b \in \mathbb{R}^{B \times d_{att}}$ denote the values and queries, and for candidate database samples $\mathbf{K}_d \in \mathbb{R}^{D \times d_{att}}$ and $\mathbf{V}_d \in \mathbb{R}^{D \times d_{out}}$ denote the keys and values. A relational attention mechanism computes the weights p_j via alignment of keys and queries using a scaled dot-product attention (Vaswani et al., 2017): $n\left(\mathbf{K}_d \mathbf{Q}_b^T / \sqrt{d_{att}}\right)$, where $n()$ is a normalization function to satisfy $p_j \geq 0$ and $\sum_{j=1}^D p_j = 1$ for which we consider softmax and sparsemax. Sparsemax encourages sparsity (for principle (v)) by mapping the Euclidean projection onto the probabilistic simplex (Martins & Astudillo, 2016). Note that the relational attention mechanism does not introduce any extra trainable parameters. The final decision block simply consists of a linear mapping from a linear combination of values that results in the output y_i (e.g. logits for classification). To guide the final decision, the following loss term consisting of three components (with equal weights) is used:

$$\begin{aligned} & L(y_i, g(\mathbf{v}_i)) + L\left(y_i, g\left(\sum_{j=1}^D p_j \mathbf{v}_j^{(c)}\right)\right) \\ & + L\left(y_i, g\left(\alpha \mathbf{v}_i + (1 - \alpha) \sum_{j=1}^D p_j \mathbf{v}_j^{(c)}\right)\right). \end{aligned} \quad (1)$$

The first term is a conventional supervised learning loss that is used to impose principles (i) and (ii). The second and third terms are the matching losses. The second term is used to impose principles (i), (iii) and (iv) such that the

linear combination of prototypes should yield the correct decision. The third term is used to strengthen the principles (iii) and (iv), where the choice of $\alpha \geq 0.5$ guarantees that the input sample itself has the largest contribution in the linear combination. In our experiments we randomly sample α from [0.45, 0.5] at each iteration.²

3.3. Sparsity regularization

To increase the sparsity of the distribution of weights further, we propose an additional regularization term $L(y_i, \hat{y}_i) + \lambda_{\text{sparse}} L_{\text{sparse}}(\mathbf{p})$ in the form of entropy where:

$$L_{\text{sparse}}(\mathbf{p}) = - \sum_{i=1}^D p_i \log(p_i + \epsilon), \quad (2)$$

where ϵ is a small number for numerical stability. $L_{\text{sparse}}(\mathbf{p})$ is maximized when \mathbf{p} has only 1 non-zero value, and minimized when $p_j = 1/D$.

3.4. Confidence estimation

The prototypical learning framework provides a linear decomposition of the decision into prototypes that have known labels. Ideally, labels of the prototypes should all be the same as the labels of the input. For confusing or challenging inputs, prototypes with different labels may receive a non-zero weight. We propose to use the quantification of the match between labels of prototypes (via linear superposition of the weight label conformity) as a confidence metric to calibrate predictions:

$$C(\mathbf{p}, \mathbf{y}) = \max_m \sum_{i=1}^D p_i \cdot I(y_i = m), \quad (3)$$

where y_i is the ground truth label of the sample i and $I()$ is the indicator function. $C(\mathbf{p}, \mathbf{y})$ is equal to 1 if and only if non-zero weights p_i all belong to the same class. When the confidence is below a certain threshold the model can refrain from making a decision.

4. Experiments

4.1. Setup

We demonstrate the canonical prototypical learning framework for image, text and tabular data classification problems (see Table 1). We summarize the corresponding encoder architectures in subsequent subsections (see supplementary materials for details). Outputs of the encoders are mapped to queries, keys and values using a fully-connected layer followed by ReLU. For values, layer normalization (Lei

²Other choices such as a fixed value or a different sampling range yield similar overall performance but qualitatively result in slightly worse results for prototype quality.

Ba et al., 2016) is employed for more stable training. A fully-connected layer is used in the decision making block, yielding logits for determining the estimated class. Softmax cross entropy loss is used as $L()$. Adam optimization algorithm is employed (Kingma & Ba, 2014) with exponential learning rate decay (with parameters optimized on a validation set).

Table 1. Datasets used in experiments. Different candidate databases are sampled randomly from the training dataset at each iteration. Training database size is chosen to fit the model to the memory of a single GPU. D at inference is chosen sufficiently large to obtain high accuracy.

| Dataset | Encoder | Database size D | |
|---------------|---------|-------------------|-----------|
| | | Training | Inference |
| MNIST | ResNet | 1024 | 32768 |
| Fashion-MNIST | ResNet | 1024 | 32768 |
| CIFAR-10 | ResNet | 1024 | 32768 |
| Fruits | ResNet | 256 | 4096 |
| ISIC Melanoma | ResNet | 256 | 4096 |
| DBPedia | VDCNN | 512 | 4096 |
| Census Income | LSTM | 4096 | 15360 |

4.1.1. IMAGE DATA ENCODING

For image encoding, we use the standard ResNet model (He et al., 2016). ResNet applies $4\times$ convolutional down-sampling before average pooling in each dimension. Large image datasets (Fruits (Muresan & Oltean, 2017) and ISIC Melanoma (ISIC, 2016)) use an extra strided convolutional layer inserted for downsampling by a factor of 2, followed by max pooling for downsampling by another factor of 2.

4.1.2. TEXT DATA ENCODING

For text encoding, we use the very deep convolutional neural network (VDCNN) (Conneau et al., 2016) model. A sequence of raw characters (of 69 different values (Zhang & LeCun, 2015)) is used as the input. VDCNN uses 9 convolutional layers with max-pooling for downsampling, eventually followed by k-max pooling (Conneau et al., 2016).

4.1.3. TABULAR DATA ENCODING

For tabular data encoding, we use an LSTM model (Hochreiter & Schmidhuber, 1997) of length 4, which inputs the feature embeddings at every timestep, after applying a dropout with a rate of 0.5. Keys, queries and values are obtained from the output of the last layer, after applying a dropout with a rate of 0.5.

4.2. Results

We evaluate our method with four sets of experiments where we: (1) qualitatively demonstrate how our method enables

Table 2. Accuracy and median number of prototypes for MNIST, Fashion-MNIST, CIFAR-10, DBPedia and Adult Census Income. The accuracy on all datasets is roughly preserved despite our model relying on a small fraction of prototypes.

| | Method | Acc. % | No. of prototypes | | |
|---------------|------------------------------------|--------|-------------------|------|-------|
| | | | 50 % | 90 % | 95 % |
| MNIST | Baseline enc. | 99.70 | Entire dataset | | |
| | Softmax attn. | 99.66 | 365 | 1324 | 1648 |
| | Sparsemax attn. | 99.69 | 2 | 4 | 5 |
| | Baseline enc. | 94.74 | Entire dataset | | |
| Fashion-MNIST | Softmax attn. | 94.42 | 712 | 2320 | 2702 |
| | Sparsemax attn. | 94.42 | 4 | 10 | 11 |
| | Sparsemax attn. + sparsity reg. | 94.47 | 1 | 2 | 2 |
| | Baseline enc. | 91.97 | Entire dataset | | |
| CIFAR-10 | Softmax attn. | 91.69 | 317 | 1453 | 1898 |
| | Sparsemax attn. | 91.44 | 5 | 14 | 16 |
| | Sparsemax attn. + sparsity reg. | 91.26 | 2 | 3 | 4 |
| | Baseline enc. | 98.25 | Entire dataset | | |
| DBP. | Softmax attn. | 98.20 | 63 | 190 | 225 |
| | Sparsemax attn. | 97.74 | 2 | 4 | 4 |
| | Baseline enc. | 85.68 | Entire dataset | | |
| | Softmax attn. | 85.64 | 2263 | 9610 | 12419 |
| Income | Sparsemax attn. | 85.58 | 20 | 57 | 67 |
| | Sparsemax attn. + sparsity reg. | 85.41 | 3 | 6 | 7 |

interpretable decisions; (2) show that it can also be used to compute a confidence metric that enables one to make a trade-off between number of predictions and overall accuracy; (3) demonstrate that it can be used to detect out-of-distribution samples; and (4) showcase that our method exhibits robustness to label noise.

4.2.1. SPARSE EXPLANATIONS OF DECISIONS

Table 2 shows the accuracy and the median number of prototypes required for a particular confidence value, i.e. the minimum number of weights that add up that value. In all cases, a very small accuracy gap is observed with the baseline encoder (that is trained in conventional supervised learning framework). To achieve a sufficiently high confidence, the number of prototypes required is much lower with sparsemax attention compared to softmax attention. There is variance between datasets in the number of prototypes required as intuitively expected from the discrepancy in the degree of similarity between the intra-class samples.

To further reduce the number of prototypes, we also consider sparsity regularization. With a small decrease in performance, the number of prototypes can be reduced to just a

handful.³ Figs. 3, 4 and 5 exemplify prototypes. We observe that generally perceptually similar samples are chosen as the prototypes with the largest weights.

4.2.2. CONFIDENCE-CONTROLLED PREDICTION

By varying the threshold for the confidence metric proposed in Eq. 3, a trade-off can be obtained for what ratio of the test samples the model makes a prediction vs. the overall accuracy it obtains (see Fig. 6). The baseline accuracy can be significantly improved (towards 100% in almost all cases) by making less predictions. In general, the smaller the number of prototypes, the smaller the trade-off space. Thus, softmax attention (which normally results in more prototypes) is better suited for confidence-controlled prediction compared to sparsemax. Compared to the state-of-the-art models, our canonical framework with simple and small models shows similar accuracy by making slightly fewer predictions. For MNIST, (Wan et al., 2013) achieves 0.21% error rate, that is obtained by our framework refraining from only 0.45% of predictions. For DBpedia, (Sachan & Petuum, 2018) achieves 0.91% error, that is obtained by our framework refraining from 3% of predictions. For Adult Census Income, CatBoost (Dorogush et al., 2018) achieves around 12% error rate, that is obtained by our framework refraining from 10% of predictions.

Fig. 6 shows the results for Melanoma classification, a medical diagnosis application. By refraining from some predictions, we demonstrate unprecedentedly high AUC values without using transfer learning or highly-customized models (Haenssle et al., 2018).

Comparison to deep k-nearest neighbors. Fig. 7 shows the comparison of our method to deep k-nearest neighbors (dkNN) (Papernot & McDaniel, 2018). As can be observed from the sharper slope of the plot, our prototypical learning framework outperforms dkNN both in terms of accuracy and in terms of finding related samples.

4.2.3. OUT-OF-DISTRIBUTION SAMPLES

Fig. 8 shows that the ratio of samples above a certain confidence level decreases significantly when the test dataset deviates from the training dataset. By comparing the ratio of samples above a certain confidence level, prototypical learning with softmax attention allows detection of deviations in the input distribution.

³We observe that increasing the sparsity coefficient too much (e.g. to yield 1-2 prototypes in most cases) may decrease the quality of prototypes in some cases as the model can overfit to discriminative features that are less perceptually meaningful.

Attention-Based Prototypical Learning Towards Interpretable, Confident and Robust Deep Neural Networks

| Prototypes | Input |
|------------|--|
| | viettesia infuscata is a moth in the arctiidae family. it was described by de toulgot in 1959. it is found in madagascar. |
| 0.45 | bellulia antemediana is a moth of the micronoctuidae family. it is known from western thailand.the wingspan is about 13 mm. the forewing is brown. the hindwing is light grey with an indistinct discal spot. the underside is unicolorous grey with a small discal spot on the hindwing. |
| 0.32 | grammia favorita is a moth of the arctiidae family. it was described by neumgen in 1890. it is found in the sand hills of nebraska nevada and north-eastern colorado. the habitat consists of prairie sand dunes.the length of the forewings is about 17.7 millimetres (0.70 in). the forewings are dark brown to black dorsally with creamy buff to pinkish buff bands. the hindwings are deep pinkish red with black markings. adults are on wing from mid |
| 0.22 | opsirhina lechriodes is a species of moth of the lasiocampidae family. it is found in new south wales and victoria.the wingspan is about 40 mm.the larvae feed on eucalyptus species. |
| | carlos abraham caadas anaya (born june 7 1980 in san salvador el salvador) is a salvadoran football player who currently plays for marte soya pango in the salvadoran second division. |
| 0.42 | verdon john howell (born 16 june 1936) is a former australian rules footballer who played senior football in tasmania and in the vfl (now afl).howell played with city-south from 1953 to 1957. he was a member of that club's 1954 and 1956 ntfa premiership teams and in 1957 won city's best and fairest award.howell made his vfl debut with the st kilda football club in 1958 after being signed from tasmania in 1953. |
| 0.22 | wolfgang paul (born 25 january 1940 in olsberg province of westphalia) is a german former football player.captaining borussia dortmund to the european cup winners cup in 1966 paul got included in helmut schn's west german squad for the 1966 fifa world cup. despite this the defender never played a match for west germany and had to retire early because of the effects an injury picked up in the late 1960s had to his game. |
| 0.16 | bradley james orr (born 1 november 1982) is an english footballer who plays for toronto fc on loan from blackburn rovers as a right-back. he is the uncle of the liverpool fc full back jon flanagan. |
| | payback is a hindi thriller film directed by sachin p. karande and produced by sarosh khan. the film released on 17 december 2010 under the archangel entertainment banner. |
| 0.44 | hush... hush sweet charlotte is a 1964 american thriller film directed and produced by robert aldrich and starring bette davis olivia de havilland joseph cotten and agnes moorehead as well as mary astor in her final film.the movie was adapted for the screen by henry farrell and lukas heller from farrell's unpublished short story what ever happened to cousin charlotte? it received seven academy award nominations. |
| 0.23 | nammal is a blockbuster malayalam movie released in 2002. it is directed by the notable director kamal and produced by davic kachapalli. the movie stars jishnu siddharth renuka menon and bhavana. it is best known for its award winning song sukamaneem nilavu sung by vidhu prathap and jyotsana. the movie was shot in the campus of government engineering college thrissur and holy trinity school palakkad.other members of the cast i |
| | hazen high school is an accredited public secondary school located in rural distant community of hazen arkansas united states. the school provides comprehensive education to more than 150 students annually in grades nine through twelve. hazen high school is one of two public high schools in prairie county and is the sole high school administered by the hazen school district. |
| 0.50 | the vivien t. thomas medical arts academy (vttmaa) is a public high school located in baltimore maryland. |
| 0.33 | liceo reguona (english: reguona high school) is a chilean high school located in reguona cachapoal province chile. |
| 0.13 | iran shahr high school is a high school in yazd iran. |

Figure 3. Example inputs and corresponding prototypes for DBPedia (with sparsemax attention). While classifying the inputs as animal, athlete, film and educational institution, prototypes have very similar sentence structure, words and concepts.

| Inputs | Prototypes | | | |
|--|---|--|--|---|
| age : 21 workclass : Private fnlwgt : 93977 education : HS-grad education_num : 9 marital_status : Never-married occupation : Craft-repair relationship : Not-in-family race : White gender : Male capital_gain : 0 capital_loss : 0 hours_per_week : 40 native_country : United-States | 0.34 age : 19 workclass : Without-pay fnlwgt : 43887 education : HS-grad education_num : 9 marital_status : Never-married occupation : Farming-fishing relationship : Own-child race : White gender : Male capital_gain : 0 capital_loss : 0 hours_per_week : 10 native_country : United-States | 0.22 age : 21 workclass : Without-pay fnlwgt : 232719 education : HS-grad education_num : 9 marital_status : Never-married occupation : Farming-fishing relationship : Own-child race : White gender : Male capital_gain : 0 capital_loss : 0 hours_per_week : 40 native_country : United-States | 0.08 age : 22 workclass : ? fnlwgt : 31102 education : Some-college education_num : 10 marital_status : Never-married occupation : Craft-repair relationship : Own-child race : Black gender : Male capital_gain : 0 capital_loss : 0 hours_per_week : 4 native_country : South | 0.07 age : 18 workclass : ? fnlwgt : 261276 education : Some-college education_num : 10 marital_status : Never-married occupation : ? relationship : Own-child race : Asian-Pac-Islander gender : Female capital_gain : 0 capital_loss : 0 hours_per_week : 40 native_country : Cambodia |
| age : 51 workclass : State-gov fnlwgt : 42801 education : Masters education_num : 14 marital_status : Married-civ-spouse occupation : Prof-specialty relationship : Husband race : White gender : Male capital_gain : 0 capital_loss : 0 hours_per_week : 70 native_country : United-States | 0.25 age : 62 workclass : Private fnlwgt : 71751 education : Some-college education_num : 10 marital_status : Married-civ-spouse occupation : Exec-managerial relationship : Husband race : White gender : Male capital_gain : 0 capital_loss : 1977 hours_per_week : 98 native_country : United-States | 0.13 age : 58 workclass : Private fnlwgt : 19103 education : Prof-school education_num : 15 marital_status : Married-civ-spouse occupation : Prof-specialty relationship : Husband race : White gender : Male capital_gain : 0 capital_loss : 0 hours_per_week : 99 native_country : United-States | 0.09 age : 64 workclass : Private fnlwgt : 98586 education : Doctorate education_num : 16 marital_status : Married-civ-spouse occupation : Prof-specialty relationship : Husband race : White gender : Male capital_gain : 0 capital_loss : 0 hours_per_week : 60 native_country : United-States | 0.09 age : 69 workclass : State-gov fnlwgt : 203072 education : Doctorate education_num : 16 marital_status : Married-civ-spouse occupation : Prof-specialty relationship : Husband race : White gender : Male capital_gain : 0 capital_loss : 0 hours_per_week : 50 native_country : United-States |

Figure 4. Example inputs and corresponding prototypes for Adult Census Income (with sparsemax attention and sparsity regularization). For the first example, all prototypes have similar age, two share similar education level and one has the same occupation. For the second example, three prototypes have the same occupation, all work more than 40 hours/week, and three have postgraduate education.

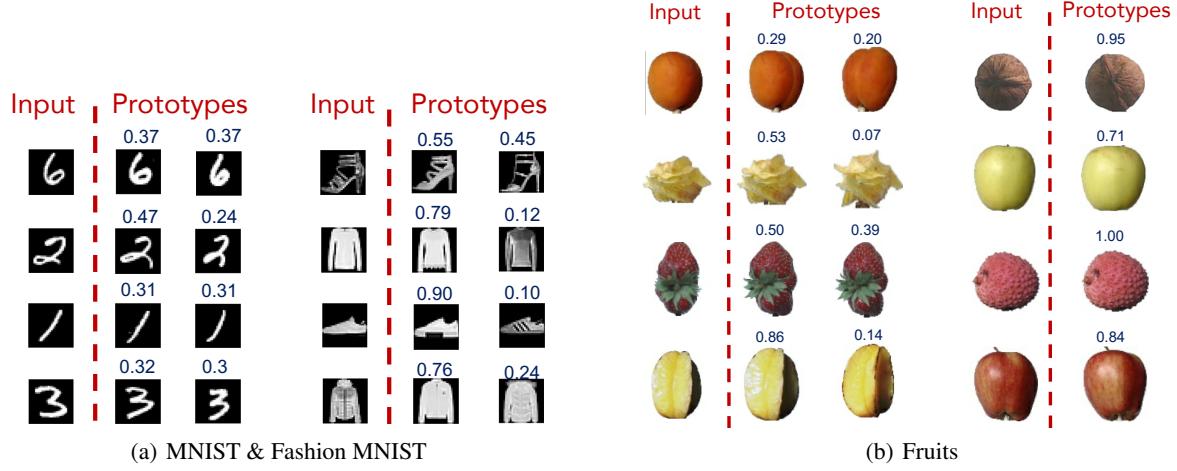


Figure 5. Example inputs and corresponding prototypes for MNIST (with sparsemax attention), Fashion-MNIST dataset (with sparsemax attention and sparsity regularization) and Fruits (with sparsemax attention and sparsity regularization). For MNIST & Fashion-MNIST, prototypes typically consist of discriminative features such as the straight line shape for the digit 1, and the long heels and strips for the sandal. For Fruits, in most cases prototypes correspond to the same fruit captured from a very similar angle.

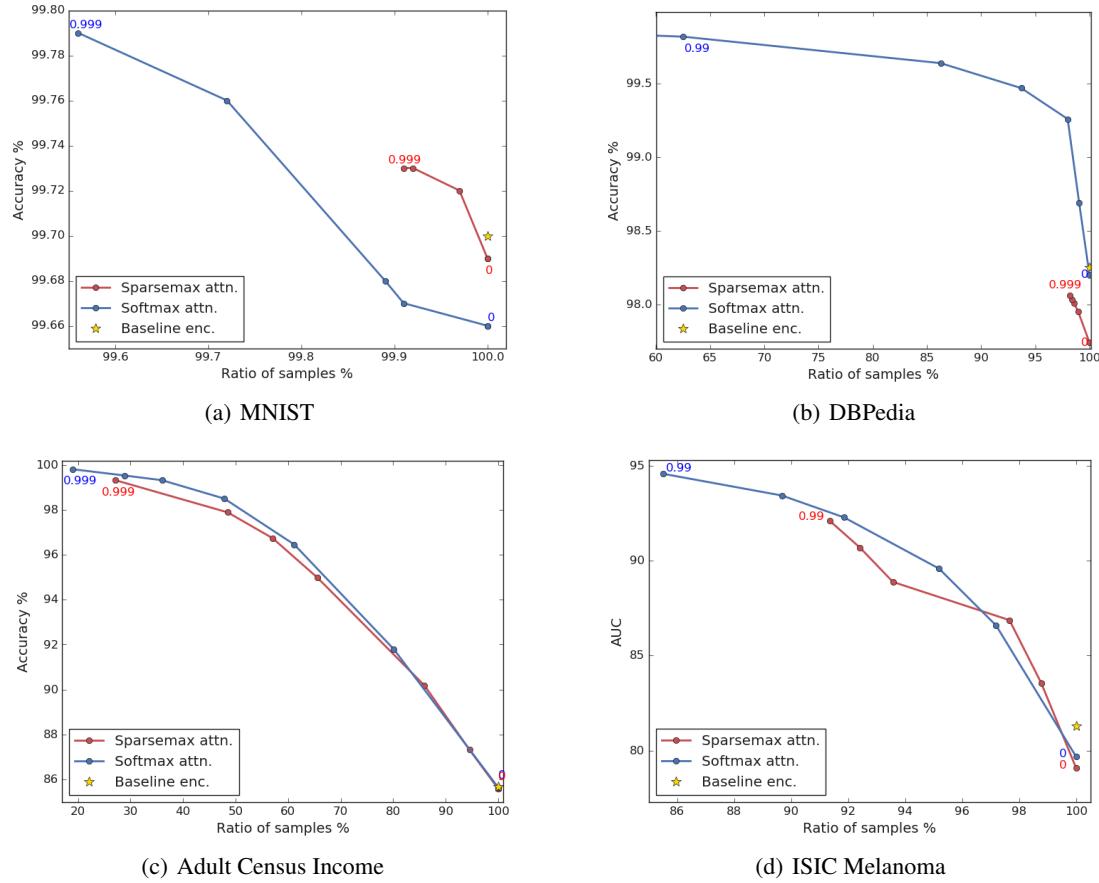


Figure 6. Accuracy vs. ratio of samples. For ISIC Melanoma, area-under-curve (AUC) is used.

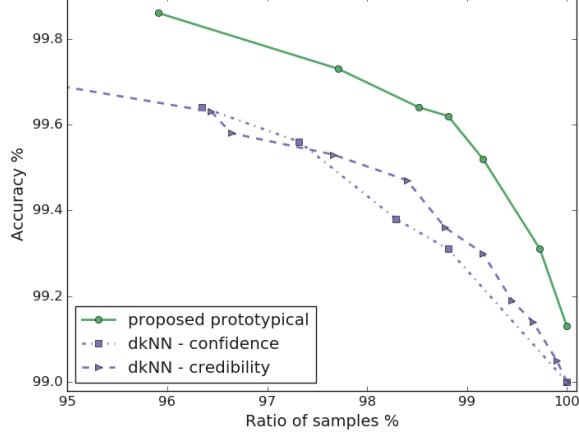


Figure 7. Comparison of dkNN (Papernot & McDaniel, 2018) and prototypical learning (with softmax attention) for MNIST. We consider both metrics in the dkNN paper, *confidence* and *credibility*. For a fair comparison we use the same network architecture as in the original dkNN paper as the encoder without data augmentation.

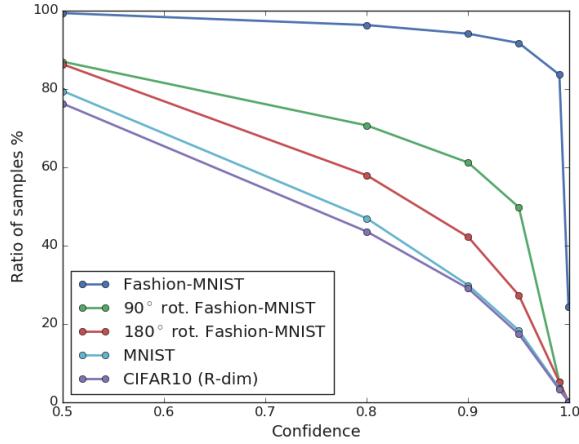


Figure 8. Ratio of samples above the confidence level for our prototypical learning model with softmax attention trained with Fashion-MNIST, and tested with rotated Fashion-MNIST, MNIST and CIFAR-10 (resizing the R dimension).

4.2.4. ROBUSTNESS TO LABEL NOISE

As prototypical learning with sparsemax attention aims to extract decision-making information from a small subset of training samples, it can be used to improve performance when the training dataset contains noisy labels. Table 3 shows results. The optimal value⁴ of λ_{sparse} increases with higher noisy label ratios, underlining the increasing importance of sparse learning.

⁴For a fair comparison, we re-optimize the learning rate parameters on a separate validation set.

Table 3. Label noise ratio vs. accuracy for baseline encoder, dropout method (Arpit et al., 2017) (optimizing the keep probability), and proposed prototypical learning model with sparsemax attention and sparsity regularization for CIFAR-10.

| Noise level | Test accuracy % | | |
|-------------|-----------------|---------|--------------|
| | Baseline | Dropout | Prototypical |
| 0.8 | 57.02 | 56.76 | 60.50 |
| 0.6 | 71.27 | 72.15 | 74.67 |
| 0.4 | 77.47 | 78.99 | 80.04 |

5. Discussion and Future Work

5.1. Computational requirements

Compared to the baseline network, the prototypical learning network only requires a very small increase in the number of learning parameters (merely two extra small matrices for the fully-connected layers to obtain queries and keys). However, it does require a longer training time and has higher memory requirements to process the candidate database. At inference, keys and values for the candidate database can be computed only once and integrated into the model. Thus, the overhead merely becomes the computation of attention outputs. During training on the other hand, both forward and backward propagation steps for the encoder need to be computed for all candidate database samples. The size of the training candidate database is limited by the memory of the processor, so in practice we sample different candidate databases randomly from the training dataset at each iteration. For faster training, data and model parallelism approaches are straightforward to implement – for example, different processors can focus on different samples of the candidate database or input batch, or they can focus on different parts of the convolution or inner product operations. Further computationally efficient approaches may involve less frequent updates for candidate queries and values.

5.2. Prototype quality

In general, the following scenarios may yield low prototype quality:

1. Lack of related samples in the candidate database,
2. Perceptual difference between humans and the encoder while focusing on discriminative features,
3. High intra-class variability that makes training difficult,
4. Imperfect encoder that cannot yield fully accurate representations of the input,
5. Insufficiency of relational attention to determine weights from queries and keys,
6. Inefficient decoupling between encoder & attention blocks and the final decision block.

There can be problem-dependent fundamental limitations on (1)-(3), whereas (4)-(6) are raised by choices of models

and losses and can be further improved. We leave the quantification of prototype quality using information-theoretic metrics or discriminative neural networks to future work.

5.3. Understanding misclassification cases



Figure 9. Example prototypes with wrong labels for CIFAR-10.

One of the benefits provided by prototypical learning is insights into wrong decision cases (as exemplified in Fig. 9). Such an analysis may yield actionable insights such as modifying the training dataset or the loss functions.

6. Conclusions

We propose an attention-based prototypical learning framework and demonstrate its usefulness for wide range of problems on image, text and tabular data. By adding a relational attention mechanism to an encoder, prototypical learning enables important novel capabilities. With sparsemax attention, it can base the learning on a few relevant samples that can be returned at inference for interpretability, and can also improve robustness to label noise. With softmax attention, it enables confidence-controlled prediction that can outperform state-of-the-art results with simple architectures by simply making slightly fewer predictions, as well as enables detecting deviations from the training data. All these are achieved while maintaining similar overall accuracy.

7. Acknowledgements

Discussions with Nicolas Papernot, Ryan Takasugi, Andrei Kouznetsov, and Andrew Moore are gratefully acknowledged.

References

- Arpit, D., Jastrzkebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., and Lacoste-Julien, S. A Closer Look at Memorization in Deep Networks. 2017.
- Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.
- Bien, J. and Tibshirani, R. Prototype selection for interpretable classification. *arXiv preprint*, arXiv:1202.5933, 2012.
- Chen, C., Li, O., Barnett, A., Su, J., and Rudin, C. This looks like that: deep learning for interpretable image recognition. *arXiv preprint*, arXiv:1806.10574, 2018.
- Conneau, A., Schwenk, H., Barrault, L., and LeCun, Y. Very deep convolutional networks for natural language processing. *arXiv preprint*, arXiv:1606.01781, 2016.
- Cui, Y., Zhou, F., Lin, Y., and Belongie, S. J. Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop. *CVPR*, 2016.
- Dorogush, A. V., Ershov, V., and Gulin, A. Catboost: gradient boosting with categorical features support. In *NIPS*, 2018.
- Erhan, D., Bengio, Y., Courville, A., and Vincent, P. Visualizing higher-layer features of a deep network. In *Technical report*, 2009.
- Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. 2017.
- Haenssle, H. A., Fink, C., Schneiderbauer, R., Toberer, F., Buhl, T., Blum, A., Kalloo, A., Hassen, A. B. H., Thomas, L., Enk, A., Uhlmann, L., study level I, R., and level II Groups. Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists. *Annals of Oncology*, 29(8):1836–1842, 2018.
- He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.
- Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
- Hoffer, E. and Ailon, N. Deep metric learning using triplet network. *arXiv preprint*, arXiv:1412.6622, 2014.
- ISIC. ISIC Archive, 2016. URL <http://www.isic-archive.com/>.
- Jerant, A. F., Johnson, J. T., Sheridan, C. D., and Caffrey, T. J. Early detection and treatment of skin cancer. *Am Fam Physician*, 2000.
- Jiang, H., Kim, B., and Gupta, M. R. To trust or not to trust a classifier. In *NIPS*, 2018.
- Kendall, A. and Gal, Y. What uncertainties do we need in bayesian deep learning for computer vision? In *NIPS*, 2017.

- Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., and Sayres, R. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *ICML*, 2018.
- Kim, W., Goyal, B., Chawla, K., Lee, J., and Kwon, K. Attention-based ensemble for deep metric learning. In *ECCV*, 2018.
- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2014.
- Koh, P. W. and Liang, P. Understanding Black-box Predictions via Influence Functions. In *ICML*, 2017.
- Lei Ba, J., Kiros, J. R., and Hinton, G. E. Layer Normalization. *arXiv preprint*, arXiv:1607.06450, 2016.
- Li, O., Liu, H., Chen, C., and Rudin, C. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. 2018.
- Martins, A. F. T. and Astudillo, R. F. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *MLR*, 2016.
- Miller, G. The magical number seven, plus or minus 2: Some limits on our capacity for processing information. *Psychological review*, 63:81–97, 04 1956.
- Mullacherry, V., Khera, A., and Husain, A. Bayesian neural networks. *arXiv preprint*, arXiv:1801.07710, 2018.
- Muresan, H. and Oltean, M. Fruit recognition from images using deep learning. *arXiv preprint*, arXiv:1712.00580, 2017.
- Papernot, N. and McDaniel, P. D. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *arXiv preprint*, arXiv:1803.04765, 2018.
- Ren, M., Liao, R., Fetaya, E., and Zemel, R. S. Incremental few-shot learning with attention attractor networks. *arXiv preprint*, arXiv:1810.07218, 2018.
- Sabour, S., Frosst, N., and Hinton, G. E. Dynamic routing between capsules. In *NIPS*, 2017.
- Sachan, D. S. and Petuum. Revisiting lstm networks for semi-supervised text classification via mixed objective function. In *KDD*, 2018.
- Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint*, arXiv:1312.6034, 2013.
- Snell, J., Swersky, K., and Zemel, R. S. Prototypical networks for few-shot learning. In *NIPS*, 2017.
- Sohn, K. Improved deep metric learning with multi-class n-pair loss objective. In *NIPS*. 2016.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. 2017.
- Vinyals, O., Blundell, C., Lillicrap, T. P., Kavukcuoglu, K., and Wierstra, D. Matching networks for one shot learning. In *NIPS*, 2016.
- Wan, L., Zeiler, M., Zhang, S., Cun, Y. L., and Fergus, R. Regularization of neural networks using dropconnect. In *ICML*, 2013.
- Yeh, C., Kim, J. S., Yen, I. E., and Ravikumar, P. Representer point selection for explaining deep neural networks. *arXiv preprint*, arXiv:1811.09720, 2018.
- Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. *arXiv preprint*, arXiv:1311.2901, 2013.
- Zhang, Q., Wu, Y. N., and Zhu, S. C. Interpretable convolutional neural networks. In *CVPR*, 2018.
- Zhang, X. and LeCun, Y. Text understanding from scratch. *arXiv preprint*, arXiv:1502.01710, 2015.

A. Training details

A.1. Image data

A.1.1. MNIST DATASET

We apply random cropping after padding each side by 2 pixels and per image standardization. The base encoder uses a standard 32 layer ResNet architecture. The number of filters is initially 16 and doubled every 5 blocks. In each block, two 3×3 convolutional layers are used to transform the input, and the transformed output is added to the input after a 1×1 convolution. $4 \times$ downsampling is applied by choosing the stride as 2 after 5th and 10th blocks. Each convolution is followed by batch normalization and ReLU nonlinearity. After the last convolution, 7×7 average pooling is applied. The output is followed by a fully-connected layer of 256 units and ReLU nonlinearity, followed by layer normalization (Lei Ba et al., 2016). Keys and queries are mapped from the output using a fully-connected layer followed by ReLU nonlinearity, where the attention size is 16. Values are mapped from the output using a fully-connected layer of 64 units and ReLU nonlinearity, followed by layer normalization. For the baseline encoder, the initial learning rate is chosen as 0.002 and exponential decay is applied with a rate of 0.9 applied every 6k iterations. The model is trained for 84k iterations. For prototypical learning model with softmax attention, the initial learning rate is chosen as 0.002 and exponential decay is applied with a rate of 0.8 applied every 8k iterations. The model is trained for 228k iterations. For prototypical learning model with sparsemax attention, the initial learning rate is chosen as 0.001 and exponential decay is applied with a rate of 0.93 applied every 6k iterations. The model is trained for 228k iterations. All models use a batch size of 128 and gradient clipping above 20.

A.1.2. FASHION-MNIST DATASET

We apply random cropping after padding each side by 2 pixels, random horizontal flipping, and per image standardization. The base encoder uses a standard 32 layer ResNet architecture, similar to our MNIST experiments. For the baseline encoder, the initial learning rate is chosen as 0.0015 and exponential decay is applied with a rate of 0.9 applied every 10k iterations. The model is trained for 332k iterations. For prototypical learning with softmax attention, the initial learning rate is chosen as 0.0007 and exponential decay is applied with a rate of 0.92 applied every 8k iterations. The model is trained for 600k iterations. For prototypical learning with sparsemax attention, the initial learning rate is chosen as 0.001 and exponential decay is applied with a rate of 0.9 applied every 8k iterations. The model is trained for 392k iterations. For prototypical learning with sparsemax attention and sparsity regularization (with $\lambda_{sparse} = 0.00008$), the initial learning rate is chosen as 0.001 and exponential decay is applied with a rate of 0.95 applied every 12k iterations. The model is trained for 450k iterations. All models use a batch size of 128 and gradient clipping above 20.

decay is applied with a rate of 0.94 applied every 8k iterations. The model is trained for 440k iterations. All models use a batch size of 128 and gradient clipping above 20.

A.1.3. CIFAR-10 DATASET

We apply random cropping after padding each side by 3 pixels, random horizontal flipping, random vertical flipping and per image standardization. The base encoder uses a standard 50 layer ResNet architecture. The number of filters is initially 16 and doubled every 8 blocks. In each block, two 3×3 convolutional layers are used to transform the input, and the transformed output is added to the input after a 1×1 convolution. $4 \times$ downsampling is applied by choosing the stride as 2 after 8th and 16th blocks. Each convolution is followed by batch normalization and the ReLU nonlinearity. After the last convolution, 8×8 average pooling is applied. The output is followed by a fully-connected layer of 256 units and the ReLU nonlinearity, followed by layer normalization (Lei Ba et al., 2016). The output is followed by a fully-connected layer of 512 units and the ReLU nonlinearity, followed by layer normalization (Lei Ba et al., 2016). Keys and queries are mapped from the output using a fully-connected layer followed by the ReLU nonlinearity, where the attention size is 16. Values are mapped from the output using a fully-connected layer of 128 units and the ReLU nonlinearity, followed by layer normalization. For the baseline encoder, the initial learning rate is chosen as 0.002 and exponential decay is applied with a rate of 0.95 applied every 10k iterations. The model is trained for 940k iterations. For prototypical learning with softmax attention, the initial learning rate is chosen as 0.0035 and exponential decay is applied with a rate of 0.95 applied every 10k iterations. The model is trained for 625k iterations. For prototypical learning with sparsemax attention, the initial learning rate is chosen as 0.0015 and exponential decay is applied with a rate of 0.95 applied every 10k iterations. The model is trained for 905k iterations. For prototypical learning with sparsemax attention and sparsity regularization (with $\lambda_{sparse} = 0.00008$), the initial learning rate is chosen as 0.0015 and exponential decay is applied with a rate of 0.95 applied every 12k iterations. The model is trained for 450k iterations. All models use a batch size of 128 and gradient clipping above 20.

CIFAR-10 experiments with noisy labels. For CIFAR-10 experiments with noisy labels for the base encoder we only optimize the learning parameters. Noisy labels are sampled uniformly from the set of labels excluding the correct one. The baseline model with noisy label ratio of 0.8 uses an initial learning rate of 0.001, decayed with a rate of 0.92 every 6k iterations, and is trained for 15k iterations. For the dropout approach, dropout with a rate of 0.1 is applied, and the model uses an initial learning rate of 0.002, decayed

with a rate of 0.85 every 8k iterations, and is trained for 24k iterations. The baseline model with noisy label ratio of 0.6 uses an initial learning rate of 0.002, decayed with a rate of 0.92 every 6k iterations, and is trained for 12k iterations. For the dropout approach, dropout with a rate of 0.3 is applied, and the model uses an initial learning rate of 0.002, decayed with a rate of 0.92 every 8k iterations, and is trained for 18k iterations. The baseline model with noisy label ratio of 0.4 uses an initial learning rate of 0.002, decayed with a rate of 0.92 every 6k iterations, and is trained for 15k iterations. For the dropout approach, dropout with a rate of 0.5 is applied, and the model uses an initial learning rate of 0.002, decayed with a rate of 0.92 every 6k iterations, and is trained for 18k iterations. For experiments for the prototypical learning model with sparsemax attention, we optimize the learning parameters and λ_{sparse} . For the model with noisy label ratio of 0.8, $\lambda_{sparse} = 0.0015$, initial learning rate is chosen as 0.0006 and exponential decay is applied with a rate of 0.95 applied every 8k iterations. The model is trained for 108k iterations. For the model with noisy label ratio of 0.6, $\lambda_{sparse} = 0.0005$, initial learning rate is chosen as 0.001 and exponential decay is applied with a rate of 0.9 applied every 8k iterations. The model is trained for 92k iterations. For the model with noisy label ratio of 0.4, $\lambda_{sparse} = 0.0003$, initial learning rate is chosen as 0.001 and exponential decay is applied with a rate of 0.9 applied every 6k iterations. The model is trained for 122k iterations.

A.1.4. FRUITS DATASET

We apply random cropping after padding each side by 5 pixels, random horizontal flipping, random vertical flipping and per image standardization. In the encoder, first, a down-sampling with a convolutional layer is applied with a stride of 2, and using 16 filters, followed by a downsampling with max-pooling with a stride of 2. After obtaining the 25×25 inputs, the a standard 32 layer ResNet architecture (similar to MNIST) is used, followed by a fully-connected layer of 128 units and the ReLU nonlinearity, followed by layer normalization (Lei Ba et al., 2016). Keys and queries are mapped from the output using a fully-connected layer followed by the ReLU nonlinearity, where the attention size is 16. Values are mapped from the output using a fully-connected layer of 64 units and the ReLU nonlinearity, followed by layer normalization. Weight decay with a factor of 0.0001 is applied for the convolutional filters. The model uses a batch size of 128 and gradient clipping above 20.

A.1.5. ISIC MELANOMA DATASET

The ISIC Melanoma dataset is formed from the ISIC Archive (ISIC, 2016) that contains over 13k dermoscopic images collected from leading clinical centers internationally and acquired from a variety of devices within each center. The dataset consists of skin images with labels de-

noting whether they contain melanoma or are benign. We construct the training and validation dataset using 15122 images (13511 benign and 1611 melanoma cases), and the evaluation dataset using 3203 images (2867 benign and 336 melanoma). While training, benign cases are undersampled in each batch to have 0.6 ratio including candidate database sets at training and inference. All images are resized to 128×128 pixels. We apply random cropping after padding each side by 8 pixels, random horizontal flipping, random vertical flipping and per image standardization. In the encoder, first, a downsampling with a convolutional layer is applied with a stride of 2, and using 16 filters, followed by a downsampling with max-pooling with a stride of 2. After obtaining the 32×32 inputs, the base encoder uses a standard 50 layer ResNet architecture (similar to CIFAR10), followed by a fully-connected layer of 128 units and the ReLU nonlinearity, followed by layer normalization (Lei Ba et al., 2016). Keys and queries are mapped from the output using a fully-connected layer followed by the ReLU nonlinearity, where the attention size is 16. Values are mapped from the output using a fully-connected layer of 64 units and the ReLU nonlinearity, followed by layer normalization. For the baseline encoder, the initial learning rate is chosen as 0.002 and exponential decay is applied with a rate of 0.9 applied every 3k iterations. The model is trained for 220k iterations. For prototypical learning with softmax attention, the initial learning rate is chosen as 0.0006 and exponential decay is applied with a rate of 0.9 applied every 3k iterations. The model is trained for 147k iterations. For prototypical learning with sparsemax attention, the initial learning rate is chosen as 0.0006 and exponential decay is applied with a rate of 0.9 applied every 4k iterations. The model is trained for 166k iterations. All models use a batch size of 128 and gradient clipping above 20.

A.2. Text data

A.2.1. DBPEDIA DATASET

There are 14 output classes: Company, Educational Institution, Artist, Athlete, Office Holder, Mean Of Transportation, Building, Natural Place, Village, Animal, Plant, Album, Film, Written Work. As the input, 16-dimensional trainable embeddings are mapped from the dictionary of 69 raw characters (Conneau et al., 2016). The maximum length is set to 448 and longer inputs are truncated while the shorter inputs are padded. The input embeddings are first transformed with a 1-D convolutional block consisting 64 filters with kernel width of 3 and stride of 2. Then, 8 convolution blocks as in (Conneau et al., 2016) are applied, with 64, 64, 128, 128, 256, 256, 512 and 512 filters respectively. All use the kernel width of 3, and after each two layers, max pooling is applied with kernel width of 3 and a stride of 2. All convolutions are followed by batch normalization and the ReLU nonlinearity. Convolutional filters use weight

normalization with parameter 0.00001. The last convolution block is followed by k-max pooling with $k=8$ (Conneau et al., 2016). Finally, we apply two fully-connected layers with 1024 hidden units. In contrast to (Conneau et al., 2016), we also use layer normalization (Lei Ba et al., 2016) after fully-connected layers as we observe this leads to more stable training behavior. Keys and queries are mapped from the output using a fully-connected layer followed by the ReLU nonlinearity, where the attention size is 16. Values are mapped from the output using a fully-connected layer of 64 units and the ReLU nonlinearity, followed by layer normalization. For the baseline encoder, initial learning rate is chosen as 0.0008 and exponential decay is applied with a rate of 0.9 applied every 8k iterations. The model is trained for 212k iterations. For prototypical learning model with softmax attention, the initial learning rate is chosen as 0.0008 and exponential decay is applied with a rate of 0.9 applied every 8k iterations. The model is trained for 146k iterations. For prototypical learning model with sparsemax attention, the initial learning rate is chosen as 0.0005 and exponential decay is applied with a rate of 0.82 applied every 8k iterations. The model is trained for 270k iterations. All models use a batch size of 128 and gradient clipping above 20. We do not apply any data augmentation.

A.3. Tabular data

A.3.1. ADULT CENSUS INCOME

There are two output classes: whether or not the annual income is above \$50k. Categorical categories such as the ‘marital-status’ are mapped to multi-hot representations. Continuous variables are used after a fixed normalization transformation. For ‘age’, the transformation first subtracts 50 and then divides by 30. For ‘fnlwgt’, the transformation first takes the log, and then subtracts 9, and then divides by 3. For ‘education-num’, the transformation first subtracts 6 and then divides by 6. For ‘hours-per-week’, the transformation first subtracts 50 and then divides by 50. For ‘capital-gain’ and ‘capital-loss’, the normalization takes the log, and then subtracts 5, and then divides by 5. The concatenated features are then mapped to a 64 dimensional vector using a fully-connected layer, followed by the ReLU nonlinearity. The base encoder uses an LSTM architecture, with 4 timesteps. At each timestep, 64-dimensional inputs are applied after a dropout with rate 0.5. The output of the last timestep is used after applying a dropout with rate 0.5. Keys and queries are mapped from this output using a fully-connected layer followed by the ReLU nonlinearity, where the attention size is 16. Values are mapped from the output using a fully-connected layer of 16 units and the ReLU nonlinearity, followed by layer normalization. For the baseline encoder, the initial learning rate is chosen as 0.002 and exponential decay is applied with a rate of 0.9 applied every 2k iterations. The model is trained for 4.5k

iterations. For the models with attention in prototypical learning framework, the initial learning rate is chosen as 0.0005 and exponential decay is applied with a rate of 0.92 applied every 2k iterations. The softmax attention model is trained for 13.5k iterations and the sparsemax attention model is trained for 11.5k iterations. For the model with sparsity regularization, the initial learning rate is 0.003 and exponential decay is applied with a rate of 0.7 applied every 2k iterations, and the model is trained for 7k iterations. All models use a batch size of 128 and gradient clipping above 20. We do not apply any data augmentation.

B. Additional results

B.1. Confidence-controlled prediction for Fashion-MNIST

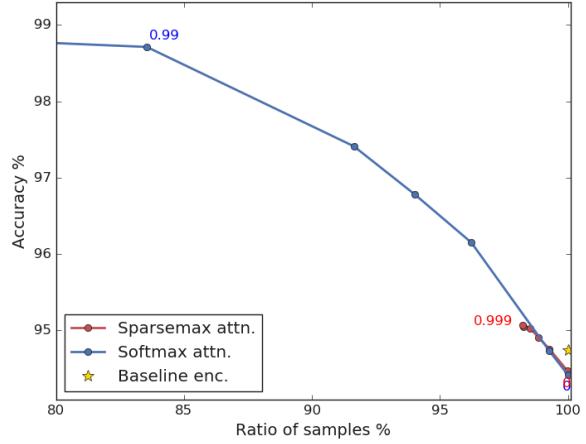


Figure 10. Fashion MNIST accuracy vs. ratio of samples, for confidence levels between 0 and 0.999.

Fig. 10 shows the accuracy vs. ratio trade-off for samples in the Fashion-MNIST dataset with sparsemax and softmax attention. The high number of prototypes enabled by softmax attention leads to a wider trade-off range.

B.2. Confidence-controlled prediction for CIFAR-10

Fig. 11 shows the accuracy vs. ratio of samples for the CIFAR-10 dataset with sparsemax and softmax attention. The high number of prototypes enabled by softmax attention leads to a wider trade-off range.

B.3. Prototype examples for CIFAR-10

Fig. 12 exemplify prototypes for CIFAR-10 with sparsemax attention, and Fig. 13 exemplify prototypes for CIFAR-10 with sparsemax attention and sparsity regularization. For most cases, we observe the discriminative features between inputs and prototypes to be similar. For examples, the shape of tires, the face patterns of dogs, the body shape of frogs,

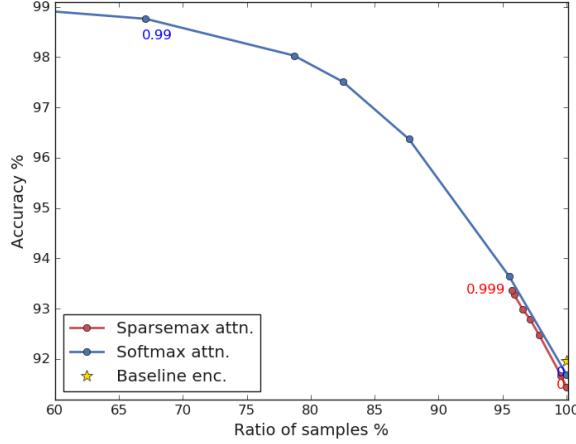


Figure 11. CIFAR-10 accuracy vs. ratio of samples for confidence levels between 0 and 0.999.

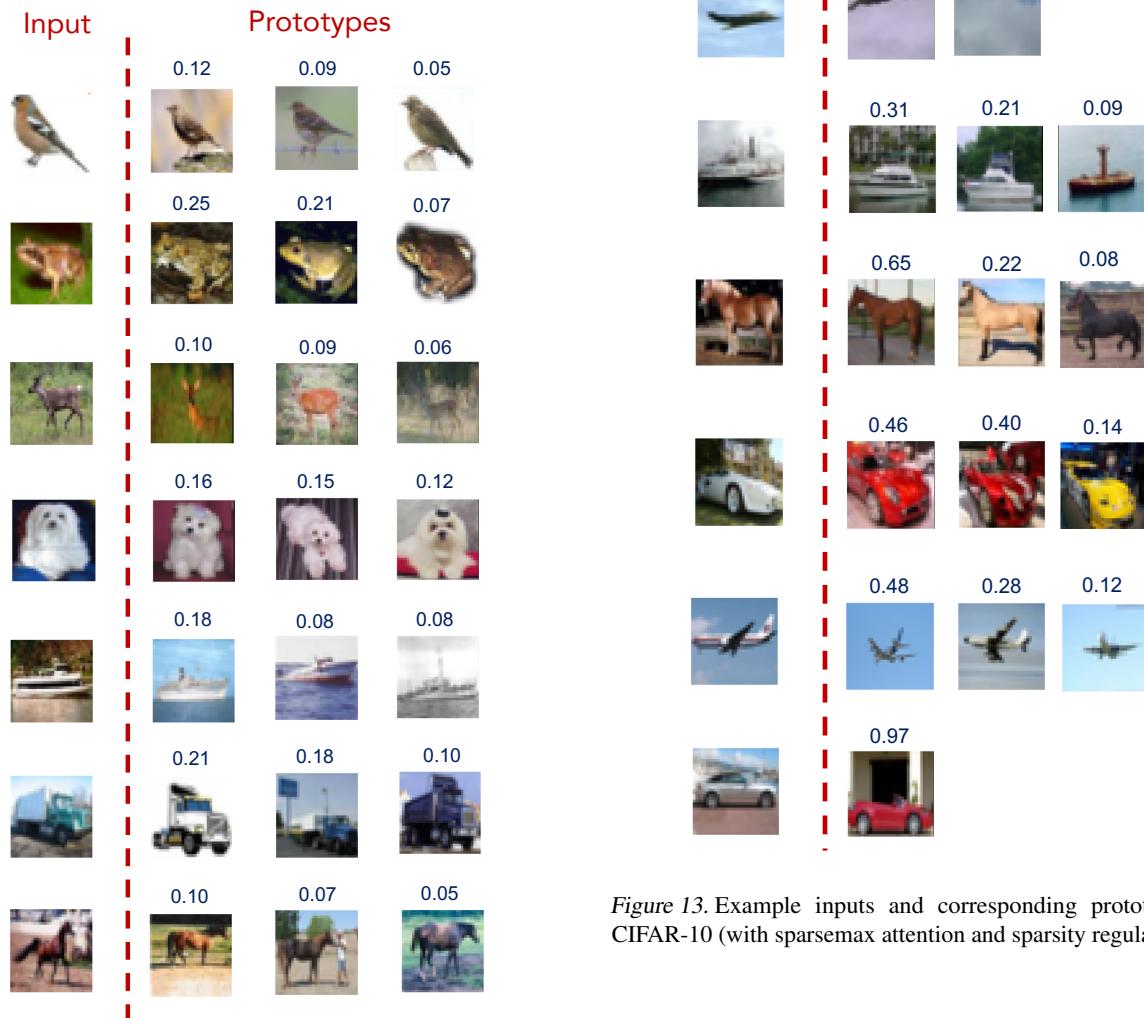


Figure 12. Example inputs and corresponding prototypes for CIFAR-10 (with sparsemax attention).

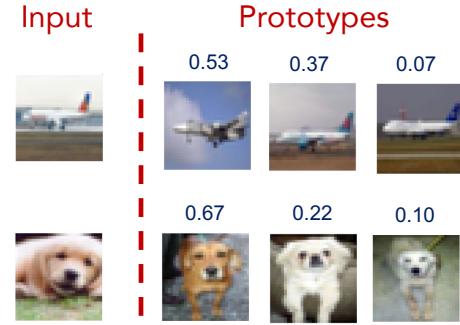


Figure 13. Example inputs and corresponding prototypes for CIFAR-10 (with sparsemax attention and sparsity regularization).

the appearance of the sky are among these features apparent in examples.

B.4. Prototype examples for ISIC Melanoma

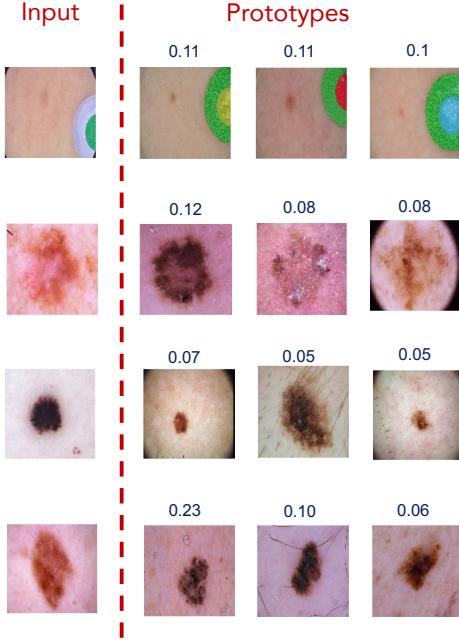


Figure 14. Example inputs and corresponding prototypes for ISIC Melanoma (with sparsemax attention).

Fig. 14 shows example prototypes for ISIC Melanoma. In some cases, we observe the commonalities between input and prototypes that distinguish melanoma cases such as the non-circular geometry or irregularly-notched borders (Jerant et al., 2000). Compared to other datasets, ISIC Melanoma dataset yields lower interpretable prototype quality on average. We hypothesize this to be due to the perceptual difficulty of the problem as well as the insufficient encoder performance shown by the lower classification accuracy (despite the acceptable AUC).