Predicting the accuracy of neural networks from final and intermediate layer outputs

Chad DeChant [1] **Seungwook Han** [1] **Hod Lipson** [2]

Abstract

We show that information about whether a neural network's output will be correct or incorrect is present in the outputs of the network's intermediate layers. To demonstrate this effect, we train a new ”meta” network to predict from either the final output of the underlying ”base” network or the output of one of the base network's intermediate layers whether the base network will be correct or incorrect for a particular input. We find that, over a wide range of tasks and base networks, the meta network can achieve accuracies ranging from 65% - 85% in making this determination.

1. Introduction

What do neural networks know and where do they know it? At what stage of a network's processing does a ”decision” get made and are there reliable markers of a correct or incorrect decision either in the output or during a network's operation at one of its intermediate layers? To begin this investigation, we ask where in a neural network's operation it becomes possible to determine whether the network might be correct or incorrect in its output for a particular input. We feed a second, ”meta” network the outputs of either an intermediate or final layer of the first, ”base”, network and train the meta network to predict whether the base network will be correct for an individual input. We call the second network a meta or metacognitive network because humans and other animals are known to make so-called metacognitive judgments to assess their confidence in the correctness of their beliefs or actions (Metcalfe et al., 1994).

We find that the meta network is able to predict whether a base network will be correct or incorrect on previously unseen inputs with up to 69% accuracy for base networks classifying ImageNet images and 85% accuracy for a base network classifying CIFAR 10 images. As these two examples suggest, the accuracy of the meta network is higher for simpler underlying tasks in our experiments.

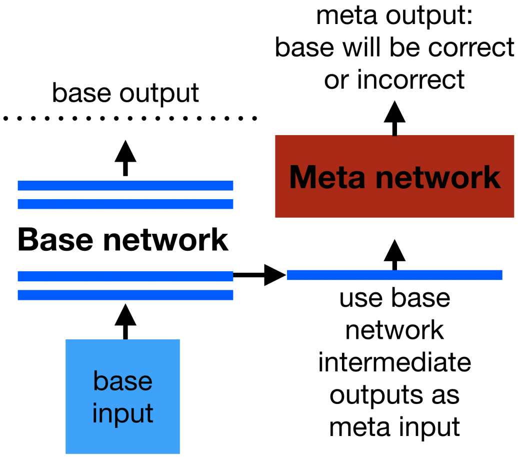

Figure 1. Meta network pipeline: the Meta network receives as input the output of one of the base network's layers for a particular input and predicts whether the base network will be correct.

The usefulness of the layers' outputs for predicting the accuracy of the network is lowest at the earliest layers in the network and increases to be highest either at the last hidden layer or, in most cases, the final output. Meta networks trained on different layers' outputs have significant but not complete overlap in which examples they are able to correctly predict will go on to be accurately or inaccurately classified, suggesting that there is slightly different information at each level which can be used to make assessments of accuracy.

2. Method

Our approach has two main stages. First, we run example images or text passages through a pretrained base network and save the final and intermediate outputs of that base network. We use PyTorch and save intermediate layer outputs using its hook feature (Paszke et al., 2017). For each example for which we save the output of intermediate stages, the

[1] Computer Science Department, Columbia University, New York, New York, USA [2] Mechanical Engineering Department and Data Science Institute, Columbia University, New York, New York, USA. Correspondence to: Chad DeChant <chad.dechant@columbia.edu>.

Identifying and Understanding Deep Learning Phenomena Workshop at the International Conference on Machine Learning 2019

Table 1. **Meta accuracy for networks on ImageNet:** Percentage accuracy of meta networks in predicting whether the base model is correct or incorrect on particular inputs when given either the final output of the base network or one of its intermediate layers' outputs. Broken out for cases when the base network was correct and incorrect. The underlying accuracy of the base model used is also given.

Model and Layer	Base Model Accuracy	Base Correct	Base Incorrect
ResNet152 Output	79	63.82	64.63
ResNet152 Last	79	63.72	63.27
DenseNet Output	77	63.61	63.40
DenseNet Last	77	61.78	62.70
VGG16 Output	74	69.23	70.07
VGG16 Last	74	68.46	69.78
VGG16 Penultimate	74	67.38	66.84
ResNet18 Output	70	67.45	69.15
ResNet18 Last	70	68.17	69.36
AlexNet Output	57	65.35	66.03
AlexNet Last	57	64.17	65.10
AlexNet Penultimate	57	63.24	63.21

base network has to be either correct or incorrect for that example. We therefore categorize these intermediate outputs as ”base correct” if the base network's eventual output is correct and ”base incorrect” if the base network's output is incorrect. Then, we train a meta classifier to predict whether the base network will be correct or incorrect from one of the intermediate or final layer outputs. Perhaps surprisingly, the meta network performs as well on previously unseen test set examples when trained on the intermediate outputs from examples which the base network was trained on as it does with intermediate outputs from examples previously unseen by the base network. We therefore train the meta network using examples from the training set of the base network.

Given the relatively high accuracy of the base models in our experiments, it would easy for the meta classifier to ”cheat” by predicting that the base network is always correct. To prevent this, we balance the classes at training time and choose our models based on the best and most balanced accuracy on the validation set. During training we define this combination of highest accuracy and balance to be the geometric mean of the meta network's accuracy on ”base correct” (C) and ”base incorrect” (I) classes minus the absolute value of their difference:

$$\sqrt[2]{C * I} - |C - I|$$

All numbers reported here are from a held out test set of inputs previously unseen by both the base and meta networks.

To determine how general and widely occurring is the phenomenon we are investigating, we train and test meta networks on a variety of tasks and base networks. Most of our testing is done on base networks which are trained for image classification tasks; we use six networks available in the PyTorch library. To assess the accuracy of networks trained on ImageNet (Russakovsky et al., 2015), we use AlexNet, Resnet 18, VGG 16, DenseNet 161, and ResNet 152 networks. For these networks we save and use for training the network's final outputs, the output of the last hidden layer (referred to as ”last” in the tables”), and in some cases the output of the penultimate hidden layer (”penultimate” in the tables).

For CIFAR 100 (Krizhevsky & Hinton, 2009) we use and train a VGG 16 network; for CIFAR 10 we train and use a VGG 19 network. For these models we train meta networks on the final output and the output of the last hidden layer, the penultimate hidden layer, the last convolutional layer, a middle convolutional layer (the fifth in VGG16, the eighth in VGG19), and the first convolutional layer. Our base networks for CIFAR 100 and CIFAR 10 had an accuracy of 71.5% and 91.1% on their respective test sets.

To test whether intermediate layers can be predictive of accuracy on a non-vision task, we use a Bi-Directional Attention Flow (BiDAF) model (Seo et al., 2016) pretrained on the Stanford Question Answering Dataset (SQuAD) version 1.1 (Rajpurkar et al., 2016). The SQuAD task gives a base network a context passage and a question, and requires the network to output where in the passage the answer to the question starts and ends. We run each example passage and question pair in the SQuAD 1.1 dataset through a pretrained model available in the AllenNLP library (Gardner et al., 2018). This base model has an exact match accuracy (where both the start and end locations of the answer predicted by the model exactly match the ground truth) of 68.03%. Further details of the BiDAF model can be found in Figure 3 in the Appendix.

3. Results

3.1. Images: ImageNet, CIFAR 100, and CIFAR 10

Accuracy numbers for the meta networks trained on various models classifying ImageNet images are found in Table 1.

For any particular layer, the meta networks display balanced accuracies when the base network as correct and incorrect, ranging from 63% to 70%.

Results for meta networks for a VGG16 network trained on CIFAR 100 can be found in Table 2; results for meta networks for a VGG19 model trained on CIFAR 10 are in Table 3.

There is a clear pattern: when a meta network is trained on the outputs of the first and middle convolutional layers, its accuracy is at best only somewhat better than chance (for CIFAR 100) and at worst no better than chance on average and very unbalanced (for CIFAR 10). Trained on the final outputs, a meta network reaches 77% accuracy averaged between the two classes for the CIFAR 100 base network and 85% for the CIFAR 10 network. Between these two extremes is a gradual increase in accuracy as meta networks are trained on later stages of the base network.

Table 2. **Meta accuracy for VGG16 on CIFAR 100:** Percentage accuracy of meta networks in predicting whether the base model is correct or incorrect on particular inputs when given either the final output of the base VGG network or one of its intermediate layers' outputs. Broken out for cases when the base VGG16 network was correct and incorrect.

Layer	Base Correct	Base Incorrect
Output	75.31	78.67
Last FC	68.42	68.56
Penultimate FC	63.77	63.57
Last Conv Layer	57.06	56.78
Middle Conv Layer	52.19	53.88
First Conv Layer	50.76	58.24

Table 3. **Meta accuracy for VGG19 on CIFAR 10:** Percentage accuracy of meta networks in predicting whether the base model is correct or incorrect on particular inputs when given either the final output of the base VGG network or one of its intermediate layers' outputs. Broken out for cases when the base VGG19 network was correct and incorrect.

Layer	Base Correct	Base Incorrect
Output	83.65	86.71
Last FC	85.76	81.31
First FC	84.52	78.15
Last Conv Layer	75.18	76.80
Middle Conv Layer	99.9	0.23
First Conv Layer	84.92	25.00

3.2. Text: SQuAD

Accuracy numbers for the meta networks trained on intermediate and final outputs of a BiDAF model for the SQuAD 1.1 dataset are found in Table 4. The output layer is the concatenation of the output prediction of the start and end locations of the answer. The pattern seen in meta network accuracies for networks trained on vision tasks is not evident here: the highest accuracy is not reached when the meta network is trained on the final outputs. Instead, the best meta network accuracy is found when classifying the output of the Modeling layer composed of Long Short Term Memory (LSTM) units just before the final output layer of the BiDAF model.

Table 4. **Meta accuracy on SQuAD BiDAF model:** Accuracy of meta networks in predicting whether the base model is correct or incorrect on particular inputs when given either the final output of the base network or one of its intermediate layers' outputs. Percentage accuracies are reported for when the base BiDAF model is entirely correct (both the start and end points of the answer are correct) and entirely incorrect (both the start and end points of the answer are incorrect) except for the last two rows which consider whether the predicted start or end points are correct independent of each other.

Layer	Base Correct	Base Incorrect
Output	66.09	68.41
Modeling	72.95	75.94
Query2Context	42.93	63.03
Context2Query	46.76	55.80
Output: start	61.92	58.11
Output: end	66.12	66.95

3.3. Overlaps between layers

We have seen that meta networks trained on the last few layers of network achieve similar accuracies. A natural question to ask about the use of many layers for a meta network, then, is whether the meta networks are getting nearly all of the same examples right even when looking at different layers' outputs. We found that while there was considerable overlap, a significant percentage (approximately 20%, depending on which layers we compare) of examples were correctly classified by a meta network looking at one layer of a VGG16 network, but not by a different meta network looking at another layer. In other words, it was not the case that the meta networks' verdicts for each example were the same no matter which layer was considered, suggesting that there might be different information about the accuracy of the base network present at different layers. Table 5 shows the overlaps of a meta network's verdicts for the outputs of the VGG16 network trained on CIFAR 100.

A similar result was evident in the meta classification results when trained on the BiDAF model for the SQuAD dataset. The overlap between correct meta network predictions of accuracy was 73.2% on examples for which the base network was correct and 75.7% for those examples which were originally incorrect coming out of the base network.

Table 5. **Overlap of meta classifications when base is correct or incorrect:** Overlaps of meta network accuracy when the base network is correct (upper right corner) and incorrect (lower left corner, in italics). Base network was a VGG16 network trained on CIFAR 100.

.

Correct Incorrect	Output	Last FC	Penultimate
Output		84.8	78.1
Last FC	*77.4*		78.9
Penultimate	*82.2*	*77.1*	

4. Discussion

It is clear that the meta networks are able to learn something about the intermediate and final outputs which are indicative of the networks' accuracy. Just what that is and whether it can be useful in improving or interpreting the networks is as yet unclear.

It is difficult to estimate the accuracy of a neural network at runtime. On tasks that involve a choice between discrete options, the value of the highest output after it is put through a softmax is often considered to represent the network's confidence or estimate of the probability of the corresponding class's being correct. However, it is not clear that this interpretation is warranted. Recent work has shown that these outputs are not reliable (Guo et al., 2017). It is interesting, then, to consider whether when a meta network is trained on the final outputs it learns to simply classify those outputs in which the predicted class has very high values as correct and those with relatively low values as incorrect. This would correspond to the general intuition that high values for predicted classes indicate meaningfully high confidence.

Figure 2 graphically illustrates the outputs of a ResNet18 network trained on ImageNet, with sample outputs of the highest confidence class arrayed along the x axis (a similar chart for outputs of the BiDAF model is found in the Appendix). It shows that while there is certainly a correlation between a base network's accuracy and the value of the output corresponding to the highest predicted class, it is not a simple or completely reliable one. On average, the base network indeed tends to be more confident in its correct answers than its wrong answers, and the set of examples the meta network is correct on shows this pattern clearly while the examples the meta network gets wrong show less distinct base "confidence" numbers. However, it is apparent that the base network is often very "confident" of a wrong answer and not confident of a correct answer. From inspecting the plots it is clear that the meta network is not judging the network's output simply by learning a threshold "confidence" level above which it predicts it will be correct and below which it predicts it will be incorrect. This is evident by the large number of incorrect high "confidence" outputs of the base network which the meta network accurately marks as incorrect, as well as the correct low "confidence" outputs which the meta networks finds correct. Further study will be required to better understand what features the meta network has learned to look for to measure accuracy.

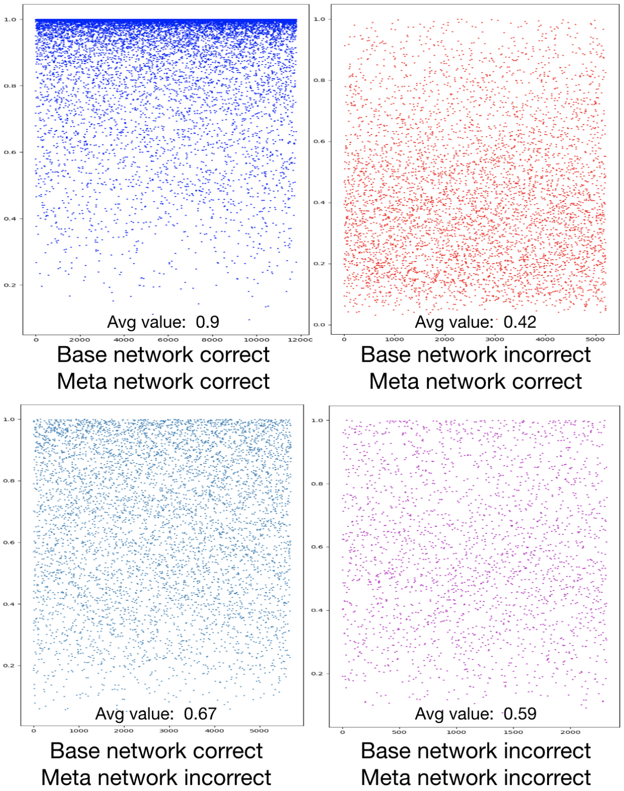

Figure 2. Examples of maximum values (arrayed along the x axis) output by a Resnet18 network on ImageNet after the softmax function. The meta network is correct in both cases in the top row and incorrect in the bottom row; the Resnet base classifier is correct on the left and incorrect on the right in both rows. The mean value in each category is given. This shows that the meta network does not learn to simply classify the output based on the value of the class prediction, which is often interpreted as the network's 'confidence'.

Neural networks designed for a classification-type task are generally trained to give an answer, not to also indicate whether they are likely to be right or wrong. While there has has certainly been work to address this, notably that involving Bayesian networks (Gal, 2016), the present work and its future extensions may point in other fruitful directions for characterizing a network's likely accuracy at runtime. There may also be interesting connections to work studying neural networks from an information theoretic perspective (Shwartz-Ziv & Tishby, 2017).

5. Acknowledgements

Supported by the US National Science Foundation National Robotics Initiative grant 1527232 and DARPA MTO grant HR0011-18-2-0020

References

Gal, Y. Uncertainty in deep learning. 2016.

Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N., Peters, M., Schmitz, M., and Zettlemoyer, L. Allennlp: A deep semantic natural language processing platform. *arXiv preprint arXiv:1803.07640*, 2018.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1321–1330. JMLR. org, 2017.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Metcalfe, J., Shimamura, A. P., et al. *Metacognition: Knowing about knowing*. MIT press, 1994.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. 2017.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Seo, M., Kembhavi, A., Farhadi, A., and Hajishirzi, H. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016.

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

A. Appendix: Additional Details

A.1. Maximum values and entropy plots from outputs of BiDAF model

The BiDAF model contains two linear output layers – each responsible for the start and end locations at which answer for the question given is predicted to lie. In the following graphs, we extend the investigation from Figure 2 to the SQuAD task and visualize the maximum values and entropies of the end output layer. The outputs were processed to convert them from logits to softmax-ed probabilities. On the one hand, these graphs are consistent with the observation from Figure 2 in the main text that the meta network is not simply learning to classify the outputs based on the value of the class prediction. On the other hand, in the vision task, the set of outputs the meta network was correct for had very different average maximum values for the predicted class and less distinguishable average values for the set of examples the meta network got wrong. However, in the NLP task, the difference in the maximum probabilities is only evident between the two classes in which the base network was correct or incorrect, regardless of the meta network.

A.2. BiDAF model

Details of the BiDAF model and a visual illustration of which intermediate layers the meta networks were trained on can be found in Figure 3

A.3. Meta network architectures

Table 1. **Meta network architecture:** Dimensions of parameters in the meta network layer by layer. FC is a fully connected layer followed in all cases but the final layer by batch normalization. Dropout applied after layers 2 and 4 in the vision meta network and layers 1, 2, 3, 4, and 5 in the NLP meta network.

Layer	Vision Meta	NLP Meta
Layer 1: FC	1024	1024
Layer 2: FC	1024	1750
Layer 3: FC	1024	512
Layer 4: FC	512	128
Layer 5: FC	512	128
Layer 6: FC	64	16
Layer 7: FC	2	2

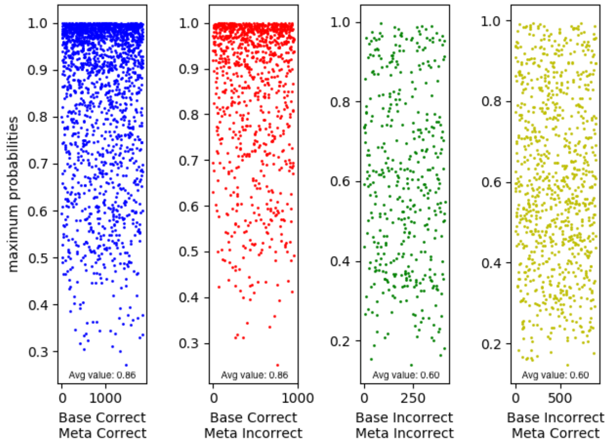

Figure 1. Examples of maximum values (arrayed along the x axis) output by the end output layer of the BiDAF model on SQuAD 1.1. The meta network's predictions on the correctness of the base network were based on the output layers (concatenated).

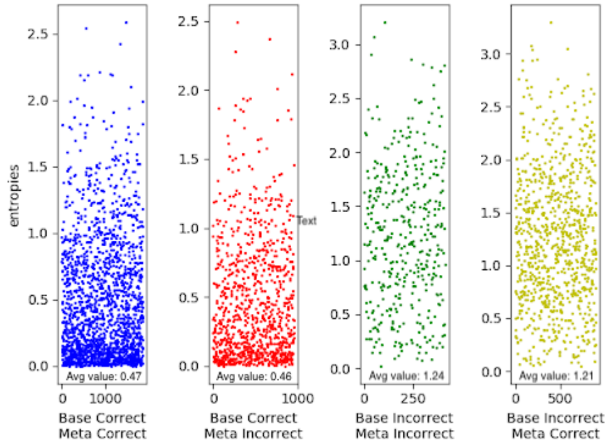

Figure 2. Examples of maximum values (arrayed along the x axis) end output by the end output layer of the BiDAF model on SQuAD 1.1 after the softmax function. The meta network's predictions on the correctness of the base network were based on the output layers (concatenated).

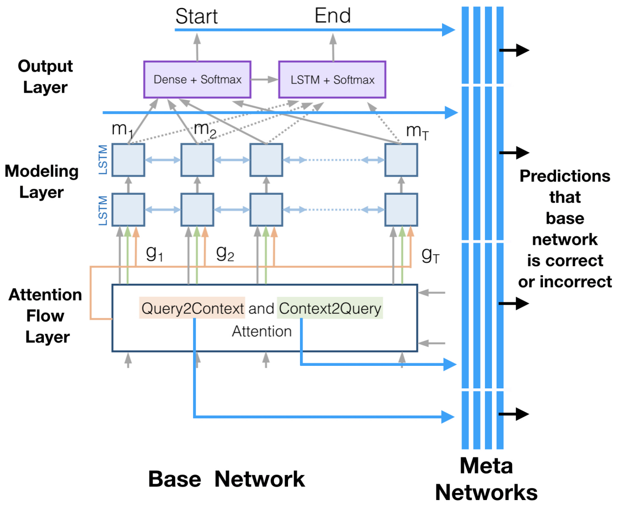

Figure 3. We train meta networks to judge whether a base network is correct or incorrect on particular inputs by feeding the meta network outputs, final or intermediate, from the base network. The blue arrows show which outputs of the base Bi-Directional Attention Flow model the meta network examines when classifying the base network's output as accurate or inaccurate. Image adapted from (Seo et al., 2016)