Finding Generalizations in Multi-Class Image Classifiers

Claudia Richoux
c@laudiacay.cool
University of Chicago

ABSTRACT

Seeing how neural nets do things, and what they learn to look for while doing that, is both important for interpretability, and also really cool. People who study thought often study "different" brains in order to better understand the general case- for example, hippocampal lesions for studying memory, or Freud or Oliver Sacks' case studies, or Descartes and his dogs, or Julian Jaynes' anecdotes about neurosurgery and discovering the functionality of brain regions. I thought applying this to neural nets might yield something interesting, so I mistrained a bunch of CNNs in targeted ways and then computed some statistics on their performance. I demonstrated that convolutional image recognition nets reliably discover "generalizations" between classes using SGD. Future research might include looking at the specific types of generalizations with deepdream or doing some sort of analysis over the weight distributions, looking at larger training data and more classes, looking at larger nets, or "messing up" the input in other ways.

KEYWORDS

neural networks, cognitive science, model visualization

ACM Reference Format:
Claudia Richoux. 2018. Finding Generalizations in Multi-Class Image Classifiers. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/1122445.1122456

1 PROBLEM

AI can be considered as a specialized form of technology intended to automate increasingly complex problems previously only solvable by brains (human or animal). Architectures like neural nets and convolution layers and Q-learning with experience replay specifically, take information-distribution forms similar to those found in nature. These architectures not only emulate human brains, but when trained with SGD similarly to how dopamine strengthens neuronal connections in the human brain, tend to have emergent behaviors that also emulate nature in eerie ways.

Some examples: the neurons closer to the eyes or input layers in image recognition nets tend to pick out smaller patterns and details. DeepDream techniques' output sometimes looks creepily like a blend of DMT and LSD visuals. AlphaZero progressed through a very similar series of chess openings as humans did throughout history. Word2Vec managed to learn that $king - man + woman = queen$, which implies it managed to pick a decent amount about human biological sex and historical human social hierarchies just from looking at relative word positions- what are concepts except generalizations? In my own interactions with transformer models on the TalkToTransformer website, I've been repeatedly awed and disturbed by how human-sounding the writing and poetry I could get it to output seemed, with well-chosen input and a bit of luck. Google Translate can sometimes come close to human translators' ability as far as conveying emotion goes, for certain passages and choices of language pairs.

Human babies' brains are extremely malleable, and their neuronal connections are heavily shaped by experience in localized training spurts throughout childhood. Various parts of their brains go from more or less neuron soup, to being able to recognize a puppy or the color red, to being able to write (hopefully semi-intelligible) month-late Deep Learning papers and (frankly just bad) Solidity :). Neural nets also start out by encoding relatively little information, but manage to recreate huge portions of complex uniquely-human behavior just by looking at their surroundings (the training data). This is very spooky to me because sometimes I wonder if they might develop the ability to experience things or have intentionality or even suffer, but that's beside the point.

In my opinion, one of the things I do that makes me feel most "human"- that a neural net could probably replicate with current tech- is when I generalize. The skills I learned in my math major with methodically constructing an argument were instrumental to finally getting good grades in a non-STEM class, or appreciating philosophy for the first time. Bottom-up AI models that transfer learn between different problems are arguably doing the same thing- they start by looking at cases of a subset of a problem (in my case, math), learn to solve that problem, and then are able to rapidly apply the knowledge elsewhere (in my case, Spinoza and HUM essays). Neural nets are trying to learn a distribution by looking at a hopefully-randomly-distributed set of examples from the distribution (or even maybe not, in the case of transfer learning). Being able to visualize these generalizations and where and when and how they appear in neural nets is not only interesting and provides insight about the anatomy of these nets' behavior, but also can provide some guidance about underlying human judgments, beliefs, and behaviors, because these nets seem to come out with quite a bit of human behavior that they were never explicitly programmed to demonstrate. Finally, it can demonstrate how much human behavior it is possible to derive from simple structures and a whole lot of input data, and get some hints about questions like how much of our behavior is learned versus inborn.

In this paper, I am seeking to look at generalizations made through incorrect training. By looking at people who grew up with different experiences and access to knowledge as adults, if we control for endogenous variables, we can understand the effects of those experiences and that knowledge on their adult behavior and

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

beliefs. From looking at all this, we can understand the underlying relationship between the two concepts. This is somewhat similar to psychoanalytic methodology- case studies of childhood behavior and adult dysfunction, then suddenly, bam! Electra complex. Similar methodology is possible with neural nets- mistrain them on certain concepts, see how they mess up on other concepts, and get an idea of how the two are related.

2 CONTEXT

Many psychologists and neurologists throughout history (mentioned in the introduction) have used various techniques of looking at what's going wrong to figure out an underlying mechanism by elimination. Currently, image recognition net visualization generally looks at neurons whose activations correlate with various classes to see correlations between classes, or backpropagates input images to maximize activation of neurons or neuron clusters in order to either discover what classes tend to activate those neurons together (showing shared information that contributes to the recognition of a class) or show what patterns cause "important" things to happen in the net (information patterns that bias the final result to a large extent). What I did is different from either of these- completely prevent the neural net from learning the idea of one class, and see how another classes' accuracy responds. This is analogous to studying brain lesions in neurology, or studying traumatized people's beliefs and problems in order to see how certain learned beliefs during upbringing impact other things.

3 APPROACH

First, I wrote a small CNN to train on the CIFAR-10 image classification task. It gets about 70-80% test accuracy on a 10-class problem after 10-15 epochs of training, so it is definitely learning all ten classes.

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 30, 30, 32)	896
max_pooling2d_4 (MaxPooling2	(None, 15, 15, 32)	0
conv2d_7 (Conv2D)	(None, 13, 13, 64)	18496
max_pooling2d_5 (MaxPooling2	(None, 6, 6, 64)	0
conv2d_8 (Conv2D)	(None, 4, 4, 64)	36928
flatten_2 (Flatten)	(None, 1024)	0
dense_4 (Dense)	(None, 64)	65600
dense_5 (Dense)	(None, 10)	650

Total params: 122,570
Trainable params: 122,570
Non-trainable params: 0

Figure 1: CNN model shape.

Next, I prepared $\binom{10}{2} = 45$ experimental groups by selecting two classes out of the 10 to randomize the labels- so if the classes were 3 and 6, half the 3's would become 6's, and vice versa. This should prevent the model from learning either of the mistaken classes, and averaging over 50 trials (trained models with different randomized labels) per group, I can get a good idea of the model's performance on each of the other groups when it fails to learn other classes. The statistics I ended up working with ended up all having low variances, thankfully, so assuming a normal distribution over these statistics, 50 is good enough to get out means for comparison.

Next, I computed the per-class categorical cross-entropy loss for a target class c and pair of confused classes i, j- M_k^{ij} is the kth model trained in this group, which can be evaluated on x as a probability distribution over the classes. The statistic below is per each trial. It measures the error on a given class for a given trial.

$$\frac{1}{|TEST|} \sum_{x,y \in TEST} \delta_{yc} \log_2(M_k^{ij}(x,c)) + (1-\delta_{yc}) \log_2(1-M_k^{ij}(x,c))$$

Then I computed mean and variance of this vector over each group, so with k as index. The variances on all of these ended up being extremely small and statistically insignificant compared to the means- everything seemed to be converging to the same result of loss for each class within a group, so I used their means to represent the performance of these confused models over various classes. As a sanity check, I ensured that loss was very high on confused classes and equal between them, and that accuracy was still somewhat alright on non-confused classes regardless of this.

For the next step, I treated the models as doing a Bayesian updating sort of thing- how does knowledge of class C_T (target class) evolve with and without the prior of knowledge of class C_C (confused class)? I started by looking at performance of models on C_T binary classification on all models where C_T was correct and two other classes were incorrect. Then, I computed this with the priors that C_C was or was not one of the incorrect classes. For the next few steps of analysis, for simplicity, I'm assuming that all of these distributions of probability errors are normal- all of the variances are very skinny compared to the means, I'm no statistician, but it feels like a safe assumption!

Next, I computed Welch's t-statistic for each given C_T and C_C pair between the samples where C_C was right, and where C_C was wrong. This is to compute whether the two samples came from the same distribution or not, and with what probability they came from different distributions. Due to time constraints of looking all these numbers up in a table to get p-values, I estimated ν with one pretty conservatively chosen value- there are probably more significant results than below, but I didn't want to accidentally p-hack.

This is a measure of whether the correct or incorrect learning of C_T significantly impacts the learning of C_C.

Finally, I computed the KL-divergence between the models where C_C is wrong but C_T is correct, and all models where C_T is correct, over the probabilities of correctly predicting C_T. This predicts the bits of information gained (as a fraction of one bit, because this has been reduced down into a binary classifier task) by learning C_C or not on the distribution of answers for C_T classification. Let mCT be the event where whether an example is C_T is predicted correctly, where $2C$ is the condition that two classes are messed up, and C_X is the condition that C_X is not one of the messed up classes. Let $C_TC_C[i,j]$ be the mean probability that an example of class j is predicted as class i in the model where both classes are trained correctly, and $C_T[i,j]$ is same mean probability when only C_T is

guaranteed to be trained correctly. Then:

$$D_{KL}(P(mCT|2C, C_C, C_T)||P(mCT|2C, C_T)$$
$$= \sum_{x \in C} C_T C_C[x,t] \log_2(\frac{C_T C_C[x,t]}{C_T[x,t]})$$

This is the bits of information added by learning the class C_T to the model's classification of the class C_C.

4 RESULTS AND DISCUSSION

In figure 2, see the statistically significant results from Welch's test. Negative numbers mean that learning the second category significantly damages the learning of the first, while positive numbers mean that learning the second category significantly improves it. Welch's t-test shows that the model generalized about cars and trucks (and also cars and frogs- look at headlights vs. frog eyes!). Bird-dog significance was slightly a statistical outlier relating to the estimation of ν, as these distributions had higher degrees of freedom. Learning about both animals and man-made things tended to confuse it- they look pretty different and have very different textures. The frog-car result was probably the most exciting result for me, because I expected car-truck to be my test for whether I had achieved any sort of interesting result, but the fact that the net spotted headlights and air intakes and how they look like frogs' eyes and mouth was really exciting!

target	confused	stdevs
airplane	dog	−2.78, p < 0.01
automobile	frog	2.116, p < 0.05
automobile	truck	2.541, p < 0.02
bird	dog	2.129, p < 0.05
bird	ship	−1.99, p < 0.05
deer	airplane	−2.21, p < 0.05
deer	truck	−2.91, p < 0.01
frog	airplane	−2.23, p < 0.05
truck	deer	−2.14, p < 0.05

Figure 2: Statistically significant results from Welch's t-statistic.

KL-entropy shows that classes whose accuracies correlate can gain up to 0.025 bits of information on classifying T by not confusing C, or lose almost the same amount by classes that tend to clash. On the entropy plot, classes 2-7 are the animals- you see patches of green and purple because the net tends to correlate animals and vehicles separately. For this plot, the classes are the indices of the following list: `['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship', 'truck']`.

From all this, I learned that the neural net does generalize across categories when learning with SGD. Not only is this the case, but you can also detect "destructive interference" when a net is asked to categorize multiple disparate groups of objects. I think this might imply it has too little space to properly learn about both animals and man-made objects. I wonder if the destructive interference might go away if there were more trainable parameters so as to give "concepts" of various classes their own space to form, but not so many as to allow quick and extreme overfitting.

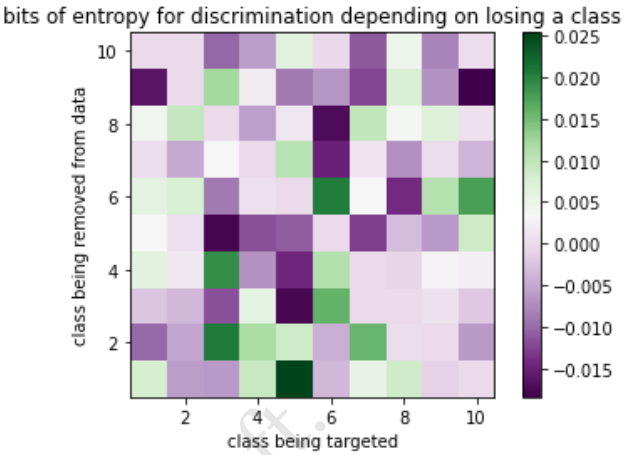

Figure 3: KL divergence for distributions of impact of mistraining one class on another. Class legend in text.

5 NEXT STEPS

There's so much more I could do with this that I didn't have time to look into. For example, it'd be interesting to mistrain at different times and then train correctly, and see how the nets recover by varying the timing of the mislabelled examples. I'd like to also look at the progressions of these statistics throughout training to detect how generalizations occur- does it slowly develop from the random noise, or do two concepts stumble onto a steep improvement by "merging" them and rapidly develop a generalization?

Deepdreaming to target various neurons and see what patterns activate them, then feeding those patterns back into the net to get the whole set of neurons targeted by a given pattern, then zeroing out the neurons above that threshold and looking at performance on various classes, could help to isolate components of a net and see how they contribute to final classification. I also thought about trying to compute some sort of dimension reduction over the earlier weights on the mistrained nets or perform kernel PCA on them or something, to try and see if I could cluster them and end up detecting which classes were mistrained based on what patterns get targeted in the earlier layers of the net.

Using these techniques on deeper models with more classes (like the imagenet task!) could be really interesting. It also might be cool to get some deeper results about how humans interact by trying this sort of thing on BERT or a sentiment classifier or something. I'm not sure how I'd manage to confuse a net on a text corpus, though, I'm unsure how I'd transform the text in a way to get cool results.

I think this methodology is really interesting, it's like the cut-rate version of Harlow's monkeys without the ethical concerns, for learning about how systems can learn to accomplish tasks that were previously only accessible to humans. Being able to chop up how a neural net learns things like this, then take apart its performance afterwards and figure out how its impacted, is something you could never ethically do with a human being to any sort of granularity, but is something you can do with an NN that models human behavior with great success. I would love to play more with this sort of thing

or see if other people do anything cool with it :)! Hoping I can write some other class papers like this during the rest of this year, I had a lot of fun with this one.

ACKNOWLEDGMENTS

My dog Alan, coffee, and friends from the lesswrong forums. :)