

Autoencoders

Michael Shell, *Member, IEEE*, John Doe, *Fellow, OSA*, and Jane Doe, *Life Fellow, IEEE*

Abstract—The abstract is None.

Index Terms—autoencoders.

I. SECTION I

SO, we have described the application of neural networks to supervised learning, in which we have labeled training examples. Now suppose we have only a set of unlabeled training examples $\{x^{(1)}, x^{(2)}, x^{(3)}, \dots\}$, where $x \in \mathbb{R}^n$. An autoencoder neural network is an unsupervised learning algorithm that applies backpropagation, setting the target values to be equal to the inputs. I.e., it uses $y^{(i)} \in x^{(i)}$. Here is an autoencoder:

The autoencoder tries to learn a function $h_{W,b}(x) \approx x$. In other words, it is trying to learn an approximation to the identity function, so as to output x that is similar to x . The identity function seems a particularly trivial function to be trying to learn; but by placing constraints on the network, such as by limiting the number of hidden units, we can discover interesting structure about the data. As a concrete example, suppose the inputs x are the pixel intensity values from a 10×10 image (100 pixels) so $n = 100$, and there are $s^2 = 50$ hidden units in layer L_2 . Note that we also have $y \in \mathbb{R}^{100}$. Since there are only 50 hidden units, the network is forced to learn a "compressed" representation of the input. I.e., given only the vector of hidden unit activations $a^{(2)} \in \mathbb{R}^{50}$, it must try to "reconstruct" the 100-pixel input x . If the input were completely random—say, each x_i comes from an IID Gaussian independent of the other features—then this compression task would be very difficult. But if there is structure in the data, for example, if some of the input features are correlated, then this algorithm will be able to discover some of those correlations. In fact, this simple autoencoder often ends up learning a low-dimensional representation very similar to PCAs. Our argument above relied on the number of hidden units s_2 being small. But even when the number of hidden units is large (perhaps even greater than the number of input pixels), we can still discover interesting structure, by imposing other constraints on the network. In particular, if we impose a "sparsity" constraint on the hidden units, then the autoencoder will still discover interesting structure in the data, even if the number of hidden units is large. Informally, we will think of a neuron as being "active" (or as "firing") if its output value is close to 1, or as being "inactive" if its output value is close to 0. We would like to constrain the neurons to be inactive most of the time. This discussion assumes a sigmoid activation function. If you are using a tanh

activation function, then we think of a neuron as being inactive when it outputs values close to -1. Recall that $a_j^{(2)}$ denotes the activation of hidden unit j in the autoencoder. However, this notation doesn't make explicit what was the input x that led to that activation. Thus, we will write $a_j^{(2)}(x)$ to denote the activation of this hidden unit when the network is given a specific input x . Further, let

$$\hat{\rho}_j = \frac{1}{m} \sum_{i=1}^m [a_j^{(2)}(x^{(i)})]$$

be the average activation of hidden unit j (averaged over the training set). We would like to (approximately) enforce the constraint

$$\hat{\rho}_j = \rho$$

where ρ is a "sparsity parameter", typically a small value close to zero (say $\rho=0.05$). In other words, we would like the average activation of each hidden neuron j to be close to 0.05 (say). To satisfy this constraint, the hidden unit's activations must mostly be near 0. To achieve this, we will add an extra penalty term to our optimization objective that penalizes $\hat{\rho}_j$ deviating significantly from ρ . Many choices of the penalty term will give reasonable results. We will choose the following:

$$\sum_{j=1}^{s_2} \rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_j}$$

Here, s_2 is the number of neurons in the hidden layer, and the index j is summing over the hidden units in our network. If you are familiar with the concept of KL divergence, this penalty term is based on it, and can also be written

$$\sum_{j=1}^{s_2} KL(\rho \parallel \text{hat}\rho_j)$$

where $KL(\rho \parallel \text{hat}\rho_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_j}$ is the Kullback-Leibler (KL) divergence between a Bernoulli random variable with mean ρ and a Bernoulli random variable with mean $\hat{\rho}_j$. KL-divergence is a standard function for measuring how different two different distributions are. (If you've not seen KL-divergence before, don't worry about it; everything you need to know about it is contained in these notes.) This penalty function has the property that $KL(\rho \parallel \text{hat}\rho_j) = 0$ if $\hat{\rho}_j = \rho$, and otherwise it increases monotonically as $\hat{\rho}_j$ diverges from ρ . For example, in the figure below, we have set $\rho=0.2$, and plotted $KL(\rho \parallel \text{hat}\rho_j)$ for a range of values of $\hat{\rho}_j$. We see that the KL-divergence reaches its minimum of 0 at

$$\hat{\rho}_j = \rho$$

M. Shell was with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332 USA e-mail: (see <http://www.michaelshell.org/contact.html>).

J. Doe and J. Doe are with Anonymous University.

Manuscript received April 19, 2005; revised August 26, 2015.

, and blows up (it actually approaches ∞) as $\hat{\rho}_j$ approaches 0 or 1. Thus, minimizing this penalty term has the effect of causing ρ_j to be close to ρ . Our overall cost function is now

$$J_{sparse}(W, b) = J(W, b) + \beta \sum_{j=1}^{s_2} KL(\rho \parallel \hat{\rho}_j)$$

mds
August 26, 2015

A. Subsection Heading Here

Subsection text here.

1) Subsubsection Heading Here: Subsubsection text here.

II. CONCLUSION

The conclusion goes here.

APPENDIX A

PROOF OF THE FIRST ZONKLAR EQUATION

Appendix one text goes here.

APPENDIX B

Appendix two text goes here.

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

- [1] H. Kopka and P. W. Daly, *A Guide to L^AT_EX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.



Michael Shell Biography text here.

John Doe Biography text here.

Jane Doe Biography text here.