(https://google.com/racialequity?authuser=0)

Training Neural Networks: Best Practices

ated Time: 5 minutes

This section explains backpropagation's failure cases and the most common way to regularize a neural network.

Failure Cases

There are a number of common ways for backpropagation to go wrong.

Vanishing Gradients

The gradients for the lower layers (closer to the input) can become very small. In deep networks, computing these gradients can involve taking the product of many small terms.

When the gradients vanish toward 0 for the lower layers, these layers train very slowly, or not at all.

The ReLU activation function can help prevent vanishing gradients.

Exploding Gradients

If the weights in a network are very large, then the gradients for the lower layers involve products of many large terms. In this case you can have exploding gradients: gradients that get too large to converge.

Batch normalization can help prevent exploding gradients, as can lowering the learning rate.

Dead ReLU Units

Once the weighted sum for a ReLU unit falls below 0, the ReLU unit can get stuck. It outputs 0 activation, contributing nothing to the network's output, and gradients can no longer flow through it during backpropagation. With a source of gradients cut off, the input to the ReLU may not ever change enough to bring the weighted sum back above 0.

Lowering the learning rate can help keep ReLU units from dying.

Dropout Regularization

Yet another form of regularization, called **Dropout**, is useful for neural networks. It works by randomly "dropping out" unit activations in a network for a single gradient step. The more you drop out, the stronger the regularization:

- 0.0 = No dropout regularization.
- 1.0 = Drop out everything. The model learns nothing.
- Values between 0.0 and 1.0 = More useful.

rms

tivation function ://developers.google.com/machine-learning/glossary?ser=0#activation_function)

opout regularization ://developers.google.com/machine-learning/glossary?ser=0#dropout_regularization)

- backpropagation (https://developers.google.com/machine-learning/glossary?authuser=0#backpropagation)
- gradient descent (https://developers.google.com/machine-learning/glossary?authuser=0#gradient_descent)

ctified linear unit (ReLU) (https://developers.google.com/machine-learning/glossary?authuser=0#ReL

Help Center (https://support.google.com/machinelearningeducation?authuser=0)

Previous

← Video Lecture (https://developers.google.com/machine-learning/crash-course/training-neural-networks/video-lecture?authuser=0)

Next

Video Lecture → (https://developers.google.com/machine-learning/crash-course/multi-class-neural-networks/video-lecture?authuser=0)