HYPER-PARAMETERS

Jay has done some great work in interactive explorations of neural networks, check out his blog.

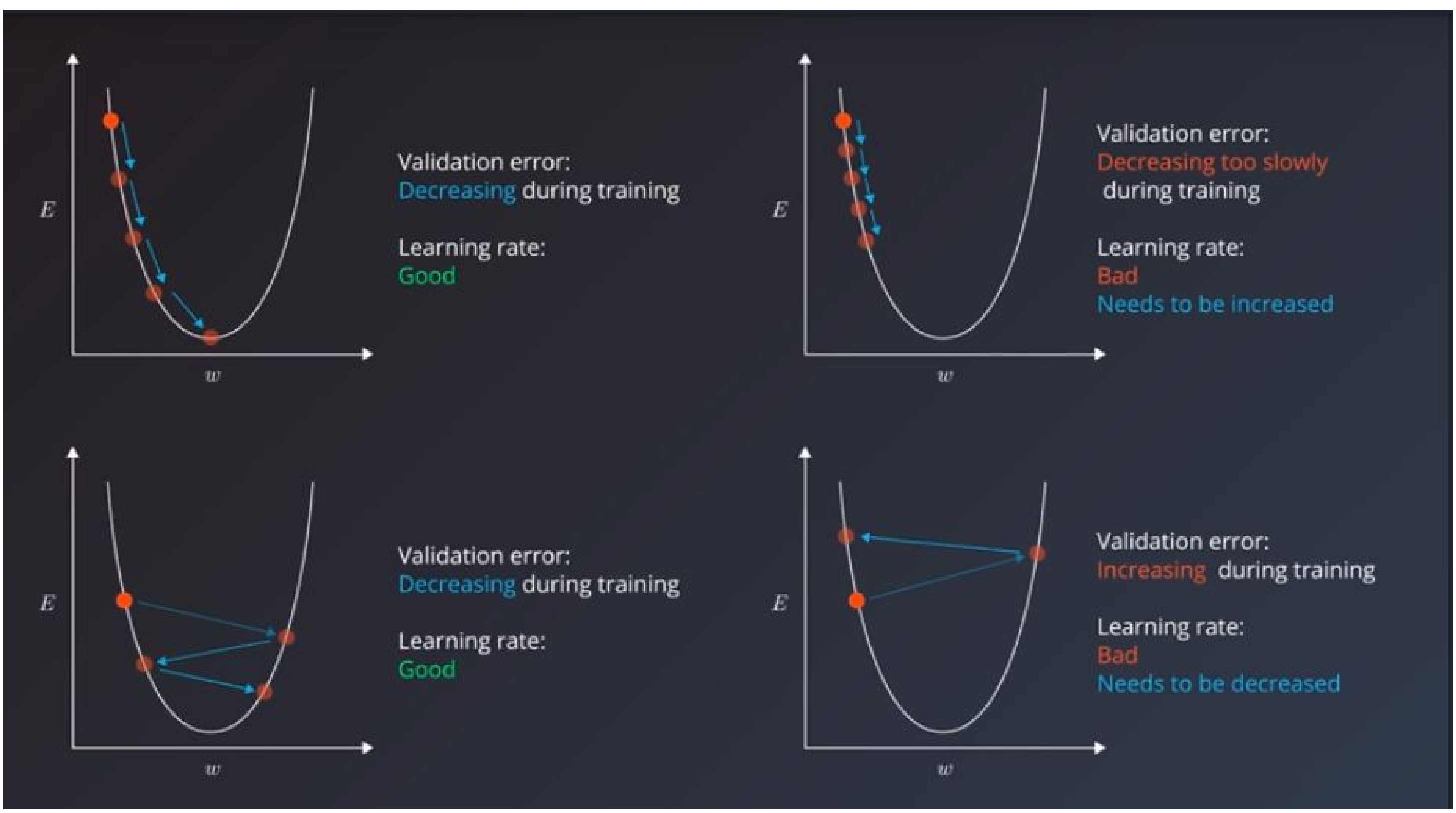
Learning Rate:

Exponential Decay in TensorFlow.

Adaptive Learning Optimizers

- AdamOptimizer
- AdagradOptimizer

Types of problems with learning rates:



MiniBatch size:

Systematic evaluation of CNN advances on the ImageNet by Dmytro Mishkin, Nikolay Sergievskiy, Jiri Matas

Epochs:

The number of training iterations is a hyperparameter we can optimize automatically using a technique called early stopping (also "early termination").

ValidationMonitor (Deprecated)

In tensorflow, we can use a ValidationMonitor with tf.contrib.learn to not only monitor the progress of training, but to also stop the training when certain conditions are met.

The following example from the ValidationMonitor documentation shows how to set it up. Note that the last three parameters indicate which metric we're optimizing.

```
validation_monitor =
tf.contrib.learn.monitors.ValidationMonitor(
  test_set.data,
  test_set.target,
  every_n_steps=50,
  metrics=validation_metrics,
  early_stopping_metric="loss",
  early_stopping_metric_minimize=True,
  early_stopping_rounds=200)
```

The last parameter indicates to ValidationMonitor that it should stop the training process if the loss did not decrease in 200 steps (rounds) of training.

The validation_monitor is then passed to tf.contrib.learn's "fit" method which runs the training process:

```
classifier = tf.contrib.learn.DNNClassifier(
  feature_columns=feature_columns,
  hidden_units=[10, 20, 10],
  n_classes=3,
  model_dir="/tmp/iris_model",
  config=tf.contrib.learn.RunConfig(save_checkpoints_secs=1))

classifier.fit(x=training_set.data,
           y=training_set.target,
           steps=2000,
           monitors=[validation_monitor])
```

SessionRunHook

More recent versions of TensorFlow deprecated monitors in favor of SessionRunHooks. SessionRunHooks are an evolving part of tf.train, and going forward appear to be the proper place where you'd implement early stopping.

At the time of writing, two pre-defined stopping monitors exist as a part of tf.train's training hooks:

- StopAtStepHook: A monitor to request the training stop after a certain number of steps

- NanTensorHook: a monitor that monitor's loss and stops training if it encounters a NaN loss

Nodes and Layers:

"in practice it is often the case that 3-layer neural networks will outperform 2-layer nets, but going even deeper (4,5,6-layer) rarely helps much more. This is in stark contrast to Convolutional Networks, where depth has been found to be an extremely important component for a good recognition system (e.g. on order of 10 learnable layers)." ~ Andrej Karpathy in https://cs231n.github.io/neural-networks-1/

More on Capacity

A more detailed discussion on a model's capacity appears in the Deep Learning book, chapter 5.2 (pages 110-120).

LSTM Vs GRU

"These results clearly indicate the advantages of the gating units over the more traditional recurrent units. Convergence is often faster, and the final solutions tend to be better. However, our results are not conclusive in comparing the LSTM and the GRU, which suggests that the choice of the type of gated recurrent unit may depend heavily on the dataset and corresponding task."

Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling by Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua Bengio

"The GRU outperformed the LSTM on all tasks with the exception of language modelling"

An Empirical Exploration of Recurrent Network Architectures by Rafal Jozefowicz, Wojciech Zaremba, Ilya Sutskever

"Our consistent finding is that depth of at least two is beneficial. However, between two and three layers our results are mixed. Additionally, the results are mixed between the LSTM and the GRU, but both significantly outperform the RNN."

Visualizing and Understanding Recurrent Networks by Andrej Karpathy, Justin Johnson, Li Fei-Fei

"Which of these variants is best? Do the differences matter? Greff, et al. (2015) do a nice comparison of popular variants, finding that they're all about the same. Jozefowicz, et al. (2015) tested more than ten thousand RNN architectures, finding some that worked better than LSTMs on certain tasks."

Understanding LSTM Networks by Chris Olah

"In our [Neural Machine Translation] experiments, LSTM cells consistently outperformed GRU cells. Since the computational bottleneck in our architecture is the softmax operation we did not observe large difference in training speed between LSTM and GRU cells. Somewhat to our surprise, we found that the vanilla decoder is unable to learn nearly as well as the gated variant."

Massive Exploration of Neural Machine Translation Architectures by Denny Britz, Anna Goldie, Minh-Thang Luong, Quoc Le

Example RNN Architectures

Application	Cell	Layers	Size	Vocabulary	Embedding Size	Learning Rate	
Speech Recognition (large vocabulary)	LSTM	5, 7	600, 1000	82K, 500K	--	--	paper
Speech Recognition	LSTM	1, 3, 5	250	--	--	0.001	paper
Machine Transla	LSTM	4	1000	Source: 160K,	1,000	--	paper

tion (seq2seq)				Target: 80K			
Image Captioning	LSTM	--	512	--	512	(fixed)	paper
Image Generation	LSTM	--	256, 400, 800	--	--	--	paper
Question Answering	LSTM	2	500	--	300	--	pdf
Text Summarization	GRU		200	Source: 119K, Target: 68K	100	0.001	pdf

SOURCES AND REFRENCES:

If you want to learn more about hyperparameters, these are some great resources on the topic:

- Practical recommendations for gradient-based training of deep architectures by Yoshua Bengio
- Deep Learning book - chapter 11.4: Selecting Hyperparameters by Ian Goodfellow, Yoshua Bengio, Aaron Courville
- Neural Networks and Deep Learning book - Chapter 3: How to choose a neural network's hyper-parameters? by Michael Nielsen
- Efficient BackProp (pdf) by Yann LeCun

More specialized sources:

- How to Generate a Good Word Embedding? by Siwei Lai, Kang Liu, Liheng Xu, Jun Zhao
- Systematic evaluation of CNN advances on the ImageNet by Dmytro Mishkin, Nikolay Sergievskiy, Jiri Matas
- Visualizing and Understanding Recurrent Networks by Andrej Karpathy, Justin Johnson, Li Fei-Fei