10-405/605 - Rectiation

Qinxin Wang

Credits: Daniel, Keshav Narayan, Siruo Zou

Carnegie Mellon University

Today’s Recitation

- SGD Recap
- Optimize SGD
- Learning Rate Tuning
- Coding Example

Carnegie Mellon University

SGD Recap

Stochastic Gradient Descent

for i in range(n):

$$w_{t+1} = w_t - \alpha * \frac{\partial F_i}{\partial w_t}$$

Gradient Descent

for i in range(n):

$$w_{t+1} = w_t - \alpha * \frac{\partial F}{\partial w_t}$$

SGD Recap

Stochastic Gradient Descent

- Computationally cheap for one step
- More steps to converge
- High variance

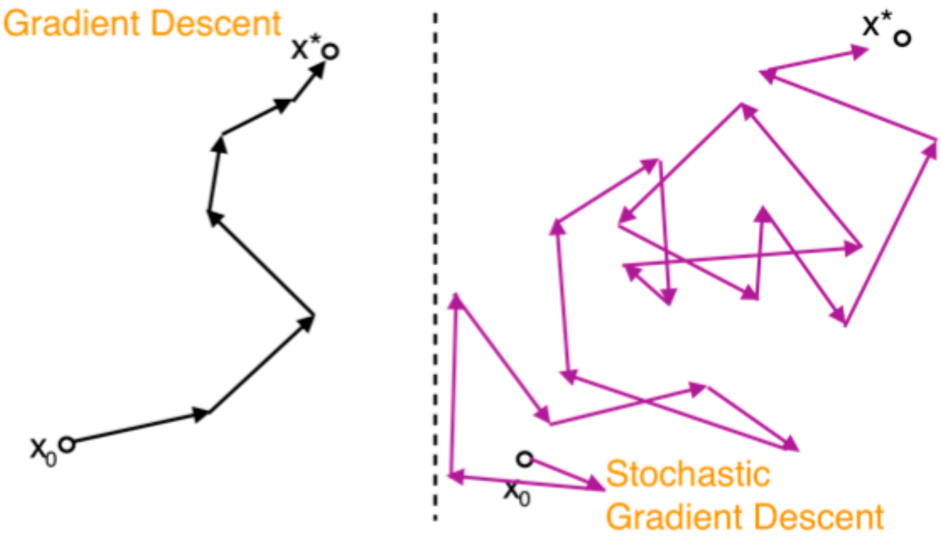
Gradient Descent

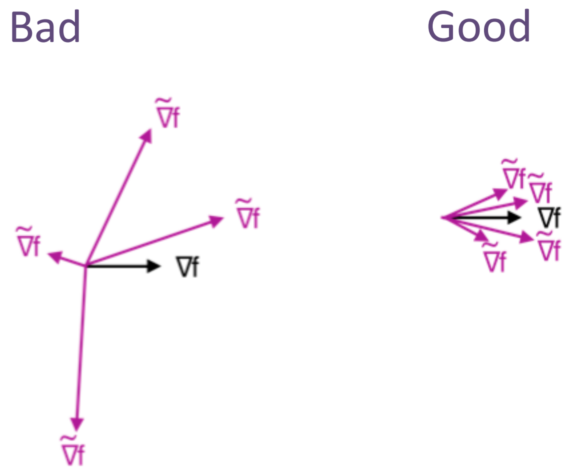
- Computationally expensive for one step
- Less steps to converge
- Low variance

In most cases, SGD can find the minimizer much faster

Carnegie Mellon University

SGD Recap





$$E\left[\left|\left|\nabla F(w_j)\right|\right|_2^2\right] \textit{ is known as the variance}$$

Credit: Yuanzhi Li, 10725 Convex Optimization

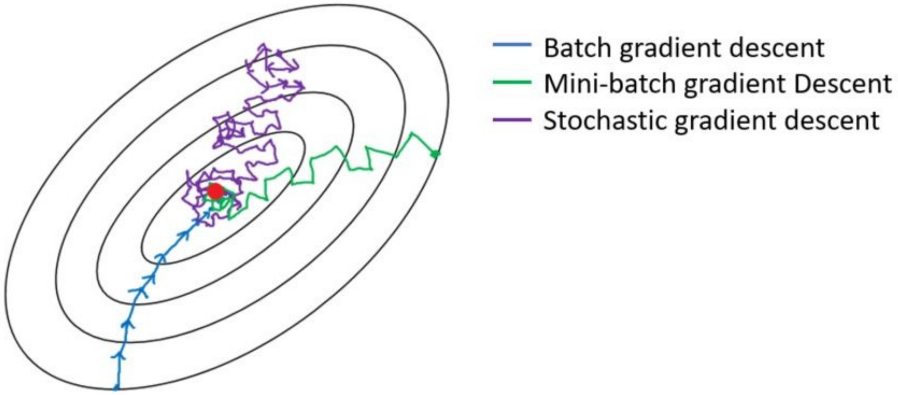
Carnegie Mellon University

SGD Recap

Mini Batch Gradient Descent

for b in batches:

$$w_{t+1} = w_t - \alpha * \frac{\partial F_b}{\partial w_t}$$



SGD with Momentum

"Noisy" derivatives for SGD.

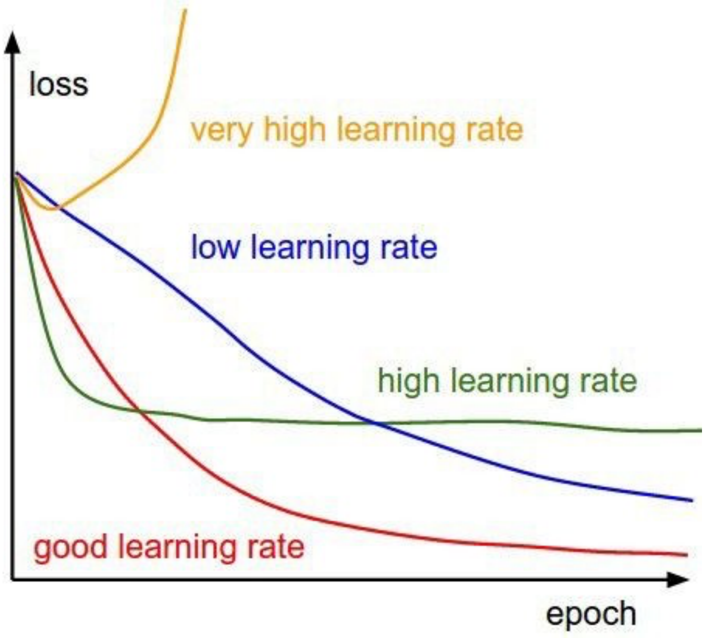
Only estimate on a small batch, which might not be the optimal direction.

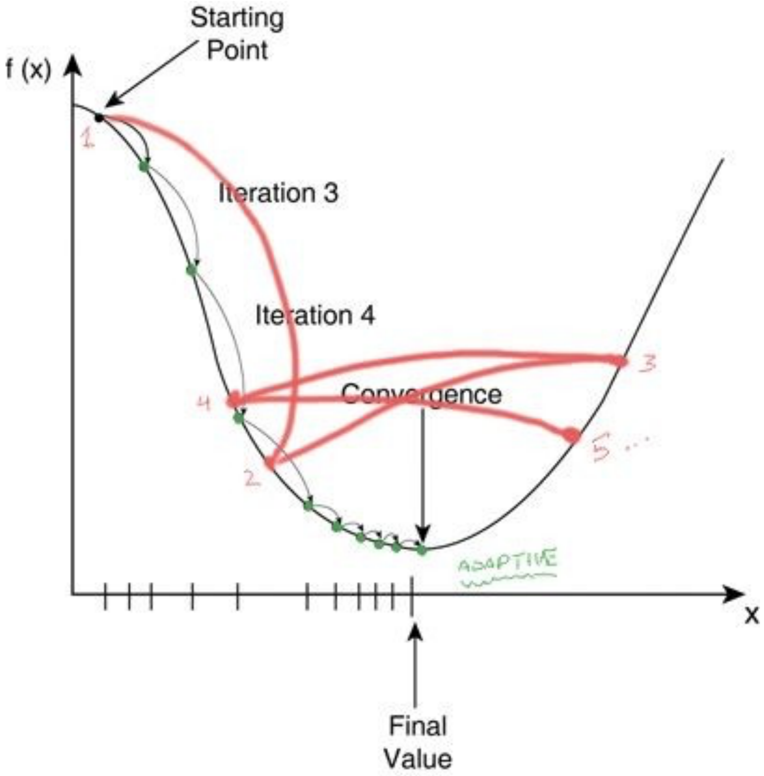
Momentum:

Define a way to get the "moving" average of some sequence, which will change along with data.

$$V_t = \beta V_{t-1} + \alpha \nabla_w L(W, X, y)$$
$$W = W - V_t$$

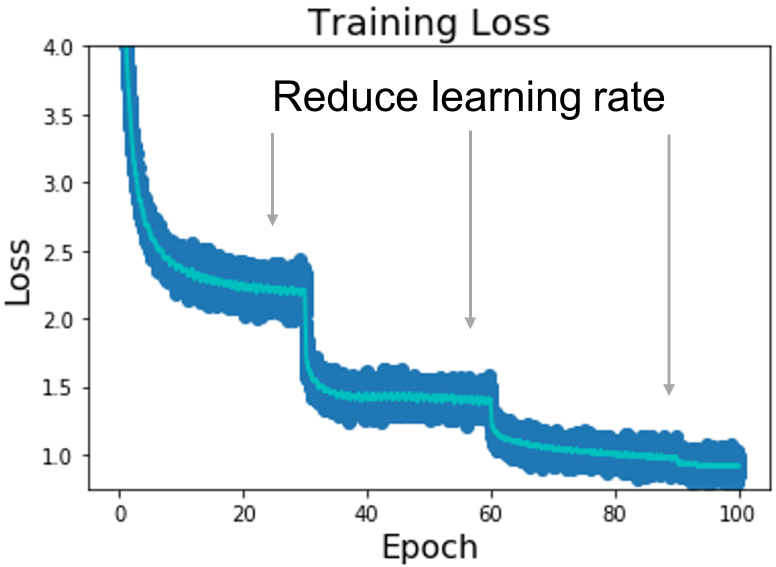
Learning Rate Tuning



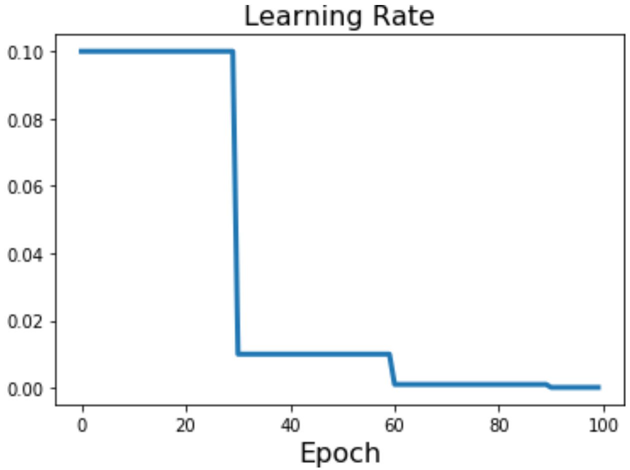


Carnegie Mellon University

Learning Rate Decay: Step



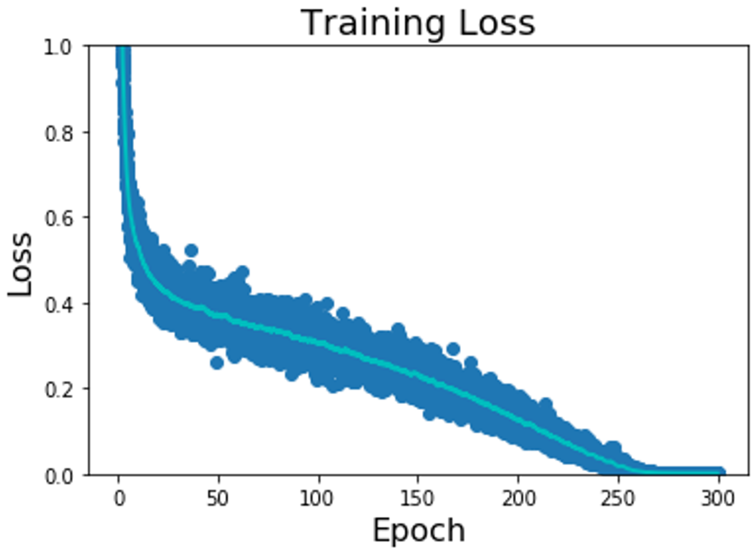
Step: Reduce learning rate at a few fixed points. E.g. for ResNets, multiply LR by 0.1 after epochs 30, 60, and 90.



Carnegie Mellon University

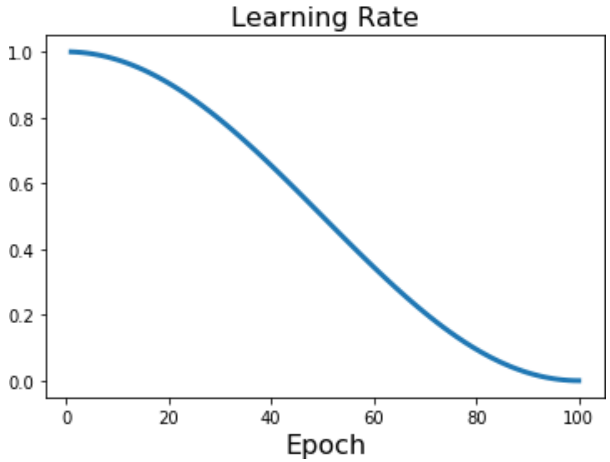
Slide Credit: Justin Johnson

Learning Rate Decay: Cosine



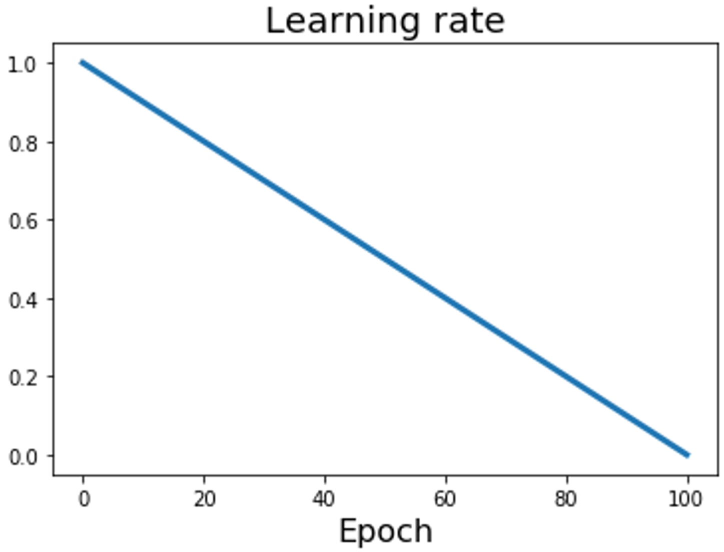
Step: Reduce learning rate at a few fixed points. E.g. for ResNets, multiply LR by 0.1 after epochs 30, 60, and 90.

Cosine: $\alpha_t = \frac{1}{2}\alpha_0 \left(1 + \cos(\frac{t\pi}{T})\right)$



Loshchilov and Hutter, "SGDR: Stochastic Gradient Descent with Warm Restarts", ICLR 2017
Radford et al, "Improving Language Understanding by Generative Pre-Training", 2018
Feichtenhofer et al, "SlowFast Networks for Video Recognition", ICCV 2019
Radosavovic et al, "On Network Design Spaces for Visual Recognition", ICCV 2019
Child at al, "Generating Long Sequences with Sparse Transformers", arXiv 2019

Learning Rate Decay: Linear



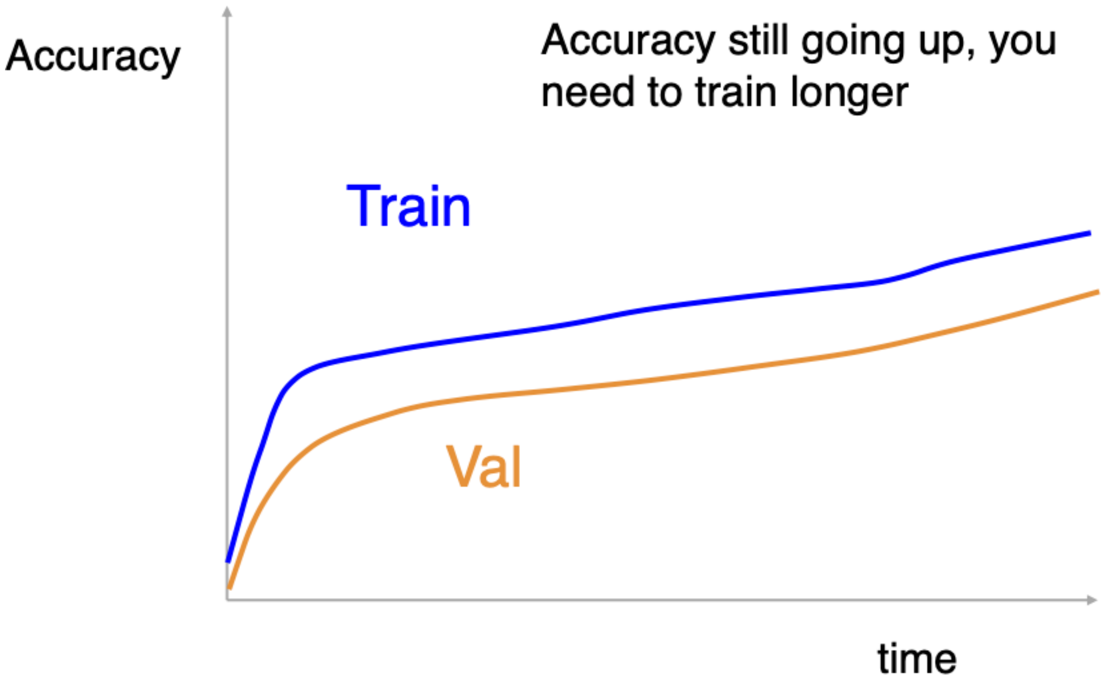
Step: Reduce learning rate at a few fixed points. E.g. for ResNets, multiply LR by 0.1 after epochs 30, 60, and 90.

Cosine: $\alpha_t = \frac{1}{2}\alpha_0\left(1 + \cos\left(\frac{t\pi}{T}\right)\right)$

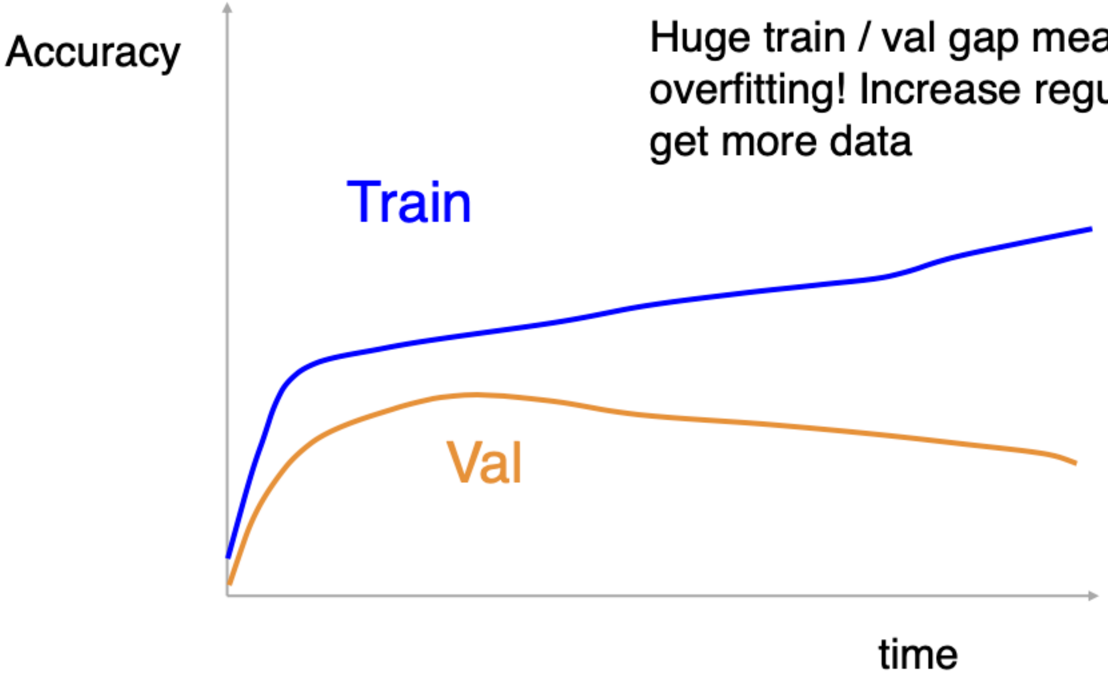
Linear: $\alpha_t = \alpha_0\left(1 - \frac{t}{T}\right)$

Devlin et al, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", NAACL 2018
Liu et al, "RoBERTa: A Robustly Optimized BERT Pretraining Approach", 2019
Yang et al, "XLNet: Generalized Autoregressive Pretraining for Language Understanding", NeurIPS 2019

Carnegie Mellon University

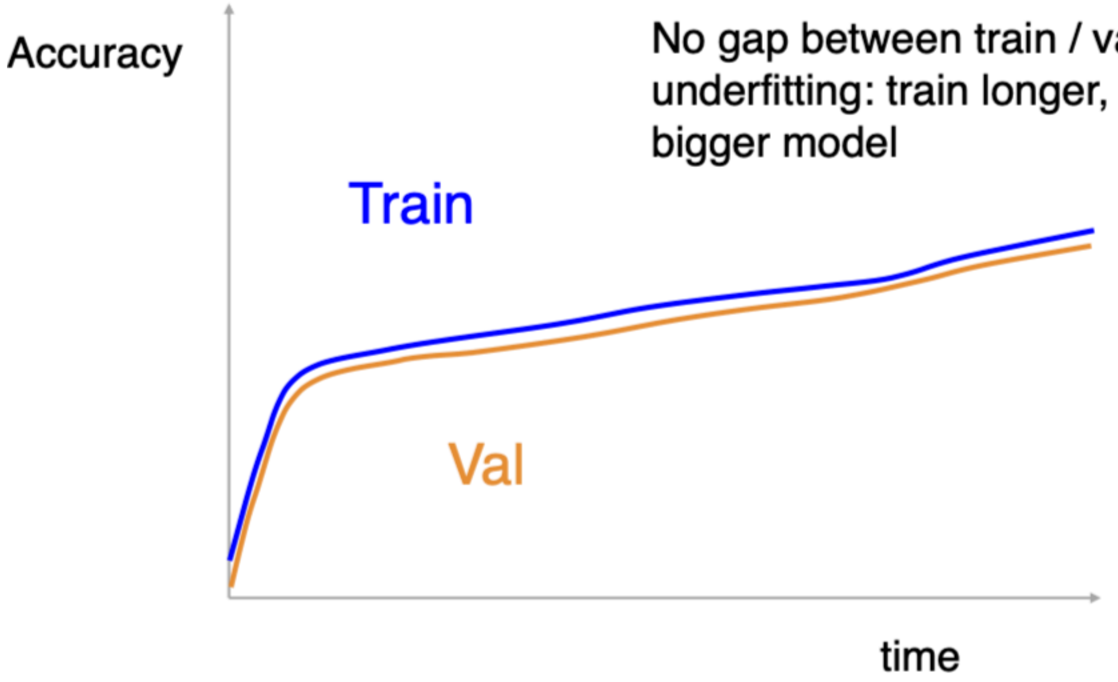


Slide Credit: Justin Johnson



Huge train / val gap means overfitting! Increase regularization, get more data

Slide Credit: Justin Johnson



No gap between train / val means underfitting: train longer, use a bigger model

Choosing Hyperparameters

Step 1: Check initial loss
Step 2: Overfit a small sample
Step 3: Find LR that makes loss go down

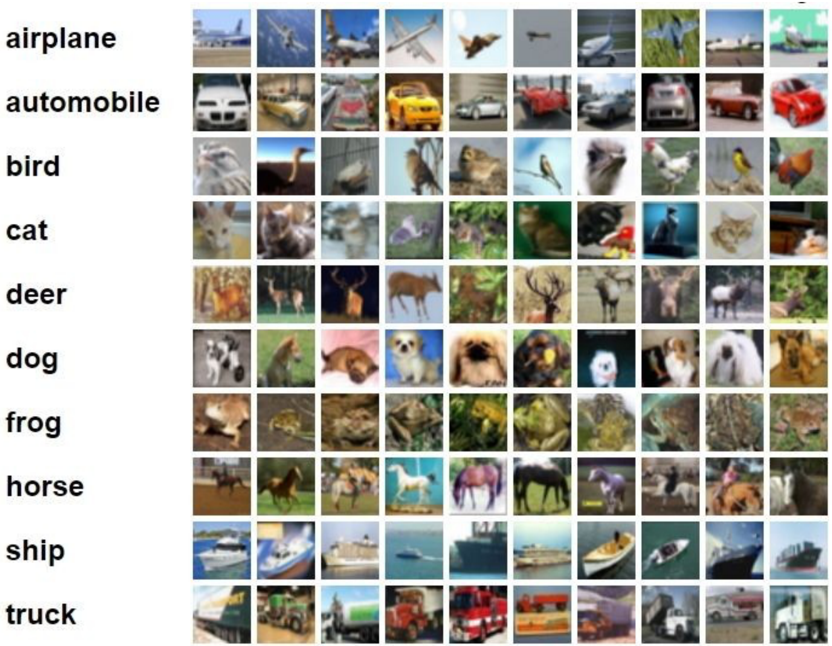
Use the architecture from the previous step, use all training data, turn on small weight decay, find a learning rate that makes the loss drop significantly within ~100 iterations

Good learning rates to try: 1e-1, 1e-2, 1e-3, 1e-4

Slide Credit: Justin Johnson

Coding Example

- Task: Image Classification
- Data: CIFAR10
- Model: CNN



Carnegie Mellon University

Reference

- https://towardsdatascience.com/https-towardsdatascience-com-why-stochastic-gradient-descent-works-9af5b9de09b8
- https://towardsdatascience.com/stochastic-gradient-descent-with-momentum-a84097641a5d
- https://deepnotes.io/sgd-momentum-adaptive#momentum

Thank you

Carnegie Mellon University