Optimization algorithms

Stochastic Gradient Descent with Momentum

- Default optimizer
  - Works well in most cases
  - Tune learning rate

for n epochs
    for B_i batches

$$\mathbf{g} := \mathbb{E}_{\mathbf{x},y \in B_i}\left[\frac{\partial \ell(\mathbf{x}, y \,|\, \theta)}{\partial \theta}\right]$$

$$\mathbf{v} := \rho \mathbf{v} + \mathbf{g}$$

$$\theta := \theta - \epsilon \mathbf{v}$$

RMSProp

- Very specialized
  - Auto-tunes learning rate
  - Momentum optional
  - Doesn't play nice with momentum
  - Works well on some reinforcement learning problems

$$\mathbf{m} := \mathbf{v} := 0$$

for n epochs
    for B_i batches

$$\mathbf{g} := \mathbb{E}_{\mathbf{x},y \in B_i}\left[\frac{\partial \ell(\mathbf{x}, y \mid \theta)}{\partial \theta}\right]$$

$$\mathbf{m} := \alpha \mathbf{m} + (1-\alpha)\mathbf{g}^2$$

$$\mathbf{v} := \rho \mathbf{v} + \frac{\mathbf{g}}{\sqrt{\mathbf{m} + \varepsilon}}$$

$$\theta := \theta - \epsilon \mathbf{v}$$

Tijmen Tieleman and Geoffrey Hinton. "Lecture 6.5-RMSProp: Divide the gradient by a running average of its recent magnitude." Neural networks for machine learning 4.2, 2012

ADAM

- Less learning rate tuning
  - Works well on small networks and problems
  - Trains well, generalizes worse
  - Mathematically not correct

$\mathbf{v} := \mathbf{m} := 0$

for n epochs

for B_i batches

$$\mathbf{g} := \mathbb{E}_{\mathbf{x},y \in B_i}[\frac{\partial \ell(\mathbf{x}, y \mid \theta)}{\partial \theta}]$$

$$\mathbf{v} := \beta_1 \mathbf{v} + (1 - \beta_1)\mathbf{g}$$

$$\mathbf{m} := \beta_2 \mathbf{m} + (1 - \beta_2)\mathbf{g}^2$$

$$s := \epsilon \frac{\sqrt{1 - \beta_2^{\text{step}}}}{1 - \beta_1^{\text{step}}}$$

$$\theta := \theta - s \frac{\mathbf{v}}{\sqrt{\mathbf{m}} + \varepsilon}$$

Kingma, D. P., & Ba, J. L.. Adam: a Method for Stochastic Optimization. ICLR 2015

What optimizer to use?

- Large models and data
    - SGD with momentum
- Small models and data
    - ADAM