Video 24

Monday, November 9, 2020 11:55 AM

Gradient descent

$$\mathcal{L}(\vec{\theta}) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}^{(i)}(\vec{\theta}) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}\left(y(x^{(i)}, \vec{\theta}), t^{(i)}\right)$$

- $\vec{\theta}$: parameters of the model (including biases and weights)
- $\mathcal{L}^{(i)}(\vec{\theta})$: loss computed over training example $(x^{(i)}, t^{(i)})$
- $y(x^{(i)}, \vec{\theta})$: model's prediction on input $x^{(i)}$ given that its parameters are set to θ.

$$\frac{\partial \mathcal{L}(\theta)}{\partial \vec{\theta}} = \frac{1}{N}\sum_{i=1}^{N} \frac{\partial \mathcal{L}^{(i)}(\vec{\theta})}{\partial \vec{\theta}}$$

by linearity

requires summing over all of the training examples.

=> batch training

Batch training is impractical when there are many training examples (when N is large).

Stochastic Gradient Descent

$$\vec{\theta} \leftarrow \vec{\theta} - \alpha \frac{\partial \mathcal{L}^{(i)}}{\partial \vec{\theta}}$$

loss for a single training example

- SGD can make progress even before it has looked at all of the training data.
- If training example is picked at random, stochastic GD is an unbiased estimate of batch GD:

$$\mathbb{E}\left[\frac{\partial \mathcal{L}^{(i)}}{\partial \vec{\theta}}\right] = \frac{1}{N}\sum_{i=1}^{N} \frac{\partial \mathcal{L}^{(i)}}{\partial \vec{\theta}} = \frac{\partial \mathcal{L}(\theta)}{\partial \vec{\theta}}$$

⚠ SGD analyses a single training example for each update, so it does not take advantage of vectorization.

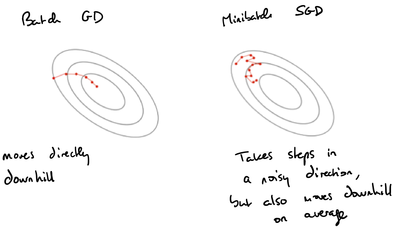
Minibatch SGD

- trade-off speed (by including more training examples in each step of SGD) with memory usage (by not including all training examples as done in batch GD).
- At each step, we sample a minibatch of training examples.
  A minibatch contains a "small" number of examples.
- Each pass through the entire training set is called an epoch
- Minibatch SGD reduces variance in the updates:

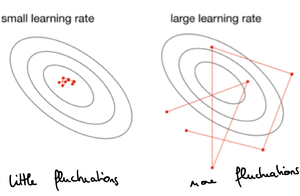
$$\mathrm{Var}\left[\frac{1}{S}\sum_{i=1}^{S} \frac{\partial \mathcal{L}^{(i)}}{\partial \vec{\theta}}\right] = \frac{1}{S^2}\mathrm{Var}\left[\sum_{i=1}^{S} \frac{\partial \mathcal{L}^{(i)}}{\partial \vec{\theta}}\right] = \frac{1}{S^2} S\,\mathrm{Var}\left[\frac{\partial \mathcal{L}^{(i)}}{\partial \vec{\theta}}\right] = \frac{1}{S}\mathrm{Var}\left[\frac{\partial \mathcal{L}^{(i)}}{\partial \vec{\theta}}\right]$$

update on a minibatch of S training examples

- Size of the minibatch (number of training examples) is a hyperparameter.
  - ↳ too large: minibatch SGD update will involve too much memory
  - ↳ too small: won't benefit from vectorization.



Influence of the learning rate:



Typical strategy:

- start with a larger learning rate
- after a few epochs, begin decreasing the learning rate.

⚠ Reducing the learning rate can reduce the loss suddenly (because there are less fluctuations in the SGD)