TTIC 31230, Fundamentals of Deep Learning

David McAllester, Winter 2020

Early Stopping, Shrinkage

Decoupled Shrinkage

Ensembles

Training Data, Validation Data and Test Data

Good performance on training data does not guarantee good performance on test data.

An nth order polynomial can fit any n (pure noise) data points.

Loss Vs. Error Rate (or BLEU Score)

While SGD is generally done on cross entropy loss, one often wants minimum classification error or BLEU Score (for translation).

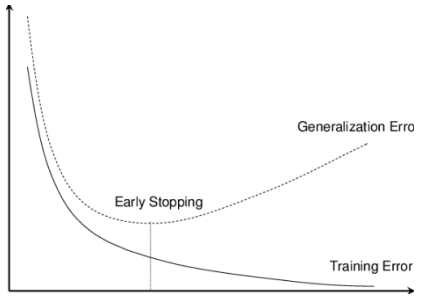
The term "loss" often refers to cross entropy loss as opposed to error rate.

SGD optimizes loss because error is not differentiable.

Later we will discuss attempts to directly optimize error.

But training on loss is generally effective.

Early Stopping



Claudia Perlich

During SGD one tracks validation loss and validation error.

One stops training when the validation error stops improving.

Empirically, loss reaches a minimum sooner than error.

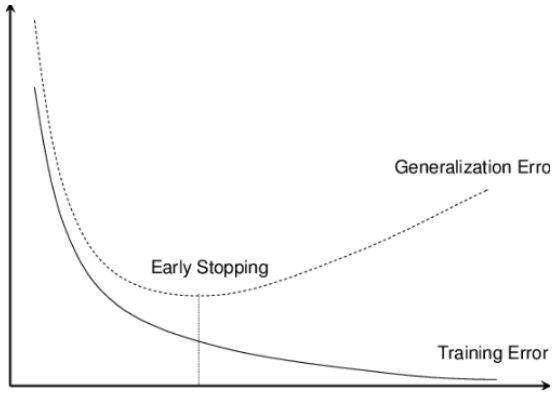
Training Data, Validation Data and Test Data

In general one designs algorithms and tunes hyper-parameters by training on training data and evaluating on validation data.

But it is possible to over-fit the validation data (validation loss becomes smaller than test loss).

Kaggle withholds test data until the final contest evaluation.

Over Confidence



Validation error is larger than training error when we stop.

The model probabilities are tuned on training data statistics.

The probabilities are tuned to an unrealistically low error rate and are therefore over-confident.

This over-confidence occurs before the stopping point and damages validation loss (as opposed to validation error).

Regularization

There is never harm in doing early stopping — one should always do early stopping.

Regularization is a modification to the training algorithm designed to reduce the training-validation gap and, in this way, improving overall performance.

Shrinkage: L_2 regularization

Will first give a Bayesian derivation. We put a prior probability on Φ and maximize the posteriori probability (MAP).

$$
\begin{aligned}
\Phi^* &= \operatorname*{argmax}_{\Phi} \; p(\Phi | \langle x_1, y_1 \rangle, \ldots, \langle x_n, y_n \rangle) \\
&= \operatorname*{argmax}_{\Phi} \; p(\Phi, \; \langle x_1, y_1 \rangle, \ldots, \langle x_n, y_n \rangle) \\
&= \operatorname*{argmax}_{\Phi} \; p(\Phi) P(\langle x_1, y_1 \rangle, \ldots, \langle x_n, y_n \rangle \mid \Phi) \\
&= \operatorname*{argmax}_{\Phi} \; p(\Phi) \prod_i \mathrm{Pop}(x_i) P_\Phi(y_i | x_i) \\
&= \operatorname*{argmax}_{\Phi} \; p(\Phi) \prod_i P_\Phi(y_i | x_i)
\end{aligned}
$$

Shrinkage: L_2 Regularization

$$\begin{aligned}\Phi^* &= \operatorname*{argmax}_{\Phi}\ p(\Phi)\ \prod_i P_\Phi(y_i|x_i) \\ &= \operatorname*{argmin}_{\Phi} \sum_i -\ln P_\Phi(y_i|x_i) - \ln p(\Phi)\end{aligned}$$

We now take a Gaussian prior

$$p(\Phi) \propto \exp\left(-\frac{||\Phi||^2}{2\sigma^2}\right)$$

Shrinkage: L_2 Regularization

$$\Phi^* = \underset{\Phi}{\operatorname{argmin}} \sum_{i=1}^{n} -\ln P_\Phi(y_i|x_i) + \frac{||\Phi||^2}{2\sigma^2}$$

$$= \underset{\Phi}{\operatorname{argmin}} \frac{1}{N} \left(\sum_{i=1}^{n} -\ln P_\Phi(y_i|x_i) + \frac{||\Phi||^2}{2\sigma^2} \right)$$

$$= \underset{\Phi}{\operatorname{argmin}} \left(E_{\langle x,\, y\rangle \sim \text{Train}} \; -\ln P_\Phi(y|x) \right) + \frac{1}{2N\sigma^2}||\Phi||^2$$

Shrinkage: L_2 Regularization

$$\nabla_\Phi \, E_{(x,y)\sim\text{Train}} \left(\mathcal{L}(\Phi, x, y) + \frac{||\Phi||^2}{2N\sigma^2} \right)$$

$$= E_{(x,y)\sim\text{Train}} \left(g(\Phi, x, y) + \frac{\Phi}{N\sigma^2} \right)$$

$$\Phi_{i+1} = \Phi_i - \eta \hat{g}_i - \frac{\eta}{N\sigma^2}\Phi$$

The last term in the update equation is called "shrinkage".

Decoupled Shrikgage

$$\Phi_{i+1} = \Phi_i - \eta\hat{g} - \frac{\eta}{N\sigma^2}\Phi_i \qquad = \Phi_i - \eta\hat{g} - \gamma\Phi_i$$

Here γ is the PyTorch shinkage parameter.

To decouple γ from other hyperparameters we can use

$$\gamma = \frac{\eta}{N_{\text{Train}}\sigma^2} = \frac{B\eta_0}{N_g N_{\text{Train}}\sigma^2}$$

where N_{Train} is the number of training instances, N_g is the decoupled momentum parameter, η_0 is the decoupled learning rate, B is the batch size, and σ is the new decoupled shrinkage parameter.

Shrinkage meets Early Stopping

Early stopping can limit $||\Phi||$.

But early stopping more directly limits $||\Phi - \Phi_{\text{init}}||$.

It seems better to take the prior on Φ to be

$$p(\Phi) \propto \exp\left(-\frac{||\Phi - \Phi_{\text{init}}||^2}{2\sigma^2}\right)$$

giving

$$\Phi_{t+1} = \Phi_t - \eta\hat{g} - \gamma(\Phi_t - \Phi_{\text{init}})$$

L_1 Regularization and Sparse Weights

$$p(\Phi) \propto e^{-\lambda||\Phi||_1} \qquad ||\Phi||_1 = \sum_i |\Phi_i|$$

$$\Phi^* = \underset{\Phi}{\text{argmin}} \quad \hat{\mathcal{L}}(\Phi) \; + \; \frac{\lambda}{N_{\text{Train}}}||\Phi||_1$$

$$\Phi \texttt{ -= } \eta \nabla_\Phi \hat{\mathcal{L}}(\Phi)$$

$$\Phi_i \texttt{ -= } (\eta\lambda/N_{\text{Train}}) \, \text{sign}(\Phi_i) \qquad \text{(shrinkage)}$$

At equilibrium (sparsity is difficult to achieve with SGD)

$$\Phi_i = 0 \qquad \text{if } |\partial\mathcal{L}/\partial\Phi_i| < \lambda/N_{\text{Train}}$$

$$\partial\mathcal{L}/\partial\Phi_i = -(\lambda/N_{\text{Train}})\text{sign}(\Phi_i) \qquad \text{otherwise}$$

Ensembles

Train several models Ens $= (\Phi_1, \ldots, \Phi_k)$ from different initializations and/or under different meta parameters.

We define the ensemble model by

$$P_{\text{Ens}}(y|x) = \frac{1}{k} \sum_{j=1}^{k} P_{\Phi_i}(y|x)$$

Ensemble models almost always perform better than any single model.

Ensembles Under Cross Entropy Loss

For log loss we average the probabilities.

$$P(y|x) = \frac{1}{k} \sum_i P_i(y|x)$$

$-\log P$ is a convex function of P. For any convex $\mathcal{L}(P)$ Jensen's inequality states that

$$\mathcal{L}\left(\frac{1}{k} \sum_i P_i\right) \leq \frac{1}{k} \sum_i \mathcal{L}(P_i)$$

This implies that the loss of the average model cannot be worse (can only be better) than the average loss of the models.

Ensembles Under Cross Entropy Loss

By Jensen:

$$\mathcal{L}\left(\frac{1}{k}\sum_i P_i\right) \leq \frac{1}{k}\sum_i \mathcal{L}(P_i)$$

However, in practice for each i we have

$$\mathcal{L}\left(\frac{1}{k}\sum_i P_i\right) \leq \mathcal{L}(P_i)$$

END