Deep Feedforward Networks - Regularization

Paolo Favaro

Workshop on Machine Learning - Observatoire de Geneve

Contents

- Regularization in Feedforward Neural Networks
  - Parameters, optimization, dataset augmentation, I/O noise, semi-supervised learning, early stopping

- Based on **Chapter 7** of Deep Learning by Goodfellow, Bengio, Courville

Regularization

- A central problem in ML is **generalization**: How do we design an algorithm that can perform well not only on training data but also on new data?

- Regularization aims at reducing the generalization error of an algorithm

Generalization

- Problems with generalization (see also Machine Learning Review slides)
  - Underfitting (large bias but low variance)
  - Overfitting (small bias but high variance)
- Neural networks typically are in the second case and regularization aims at reducing variance

Regularization

- Strategies
  - Constrain model (e.g., restrict model family or parameter space)
  - Add terms to loss function (equivalent to soft constraints to the model) — can encode priors
  - Ensemble methods (combine multiple hypotheses)

Dataset Augmentation

- The best way to make the model generalize well is to train it on more data
- One way to augment our dataset is to apply a number of realistic transformations to the data we already have and create new synthetic samples, which share the same label
- This process of data manipulation is also called **jittering**

Dataset Augmentation

original

affine distortion

horizontal flip

noise

random translation

elastic deformation

hue shift

Noise Robustness

- Apply noise to the input data at each iteration
- Apply noise to the inputs of the hidden units (Poole et al 2014)
- **Dropout** can be seen as multiplicative noise

Noise Robustness

- Apply noise to the weights
  - Model weights as random variables
  - Encourage stability of learned mapping (weights find minima with a flat neighborhood)

Label Smoothing

- Labels might be wrong (remember: it is human annotation)
- Let us model noise in the labels

labeling

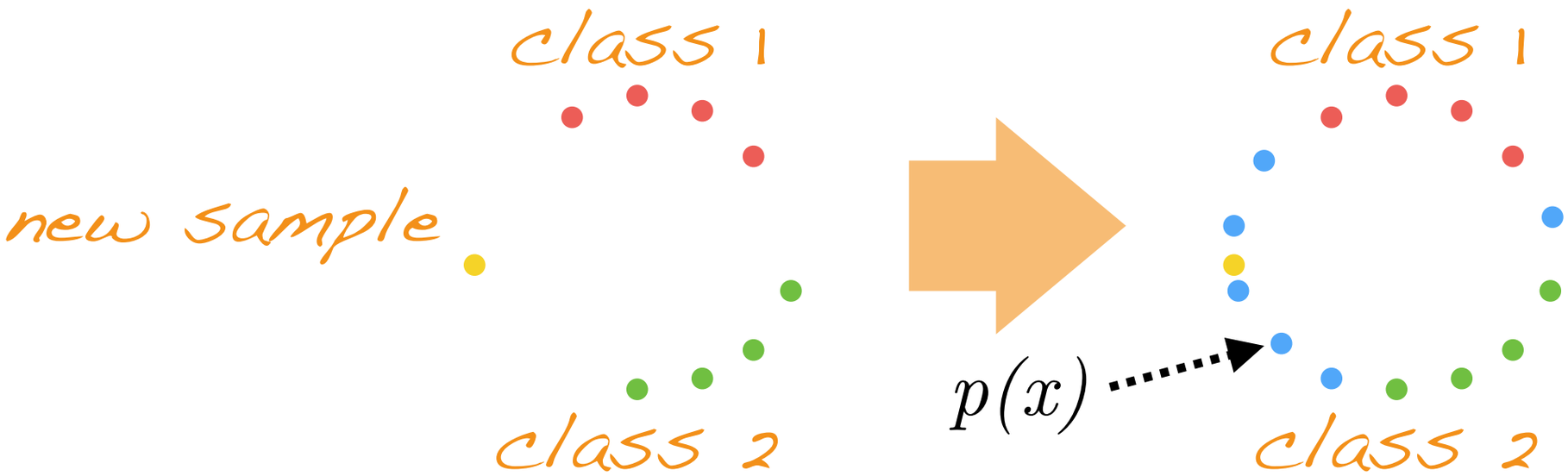
- For example, $p(y) = (1-\epsilon)\hat{p}(y) + \epsilon\mathcal{U}[1, K]$
- **Label smoothing** replaces 0s and 1s with

$$\frac{\epsilon}{K-1} \quad \text{and} \quad 1-\epsilon \quad \text{respectively}$$

Semi-Supervised Learning

- Semi-supervised learning uses unlabeled samples from $p(x)$ and labeled samples from $p(x,y)$ to build $p(y|x)$ or directly predict y from x

- The probability density $p(x)$ can be seen as a prior on the input data



Semi-Supervised Learning

- Learn a representation so that samples from the same class have similar representations

- Then a linear classifier may achieve better generalization

Early Stopping

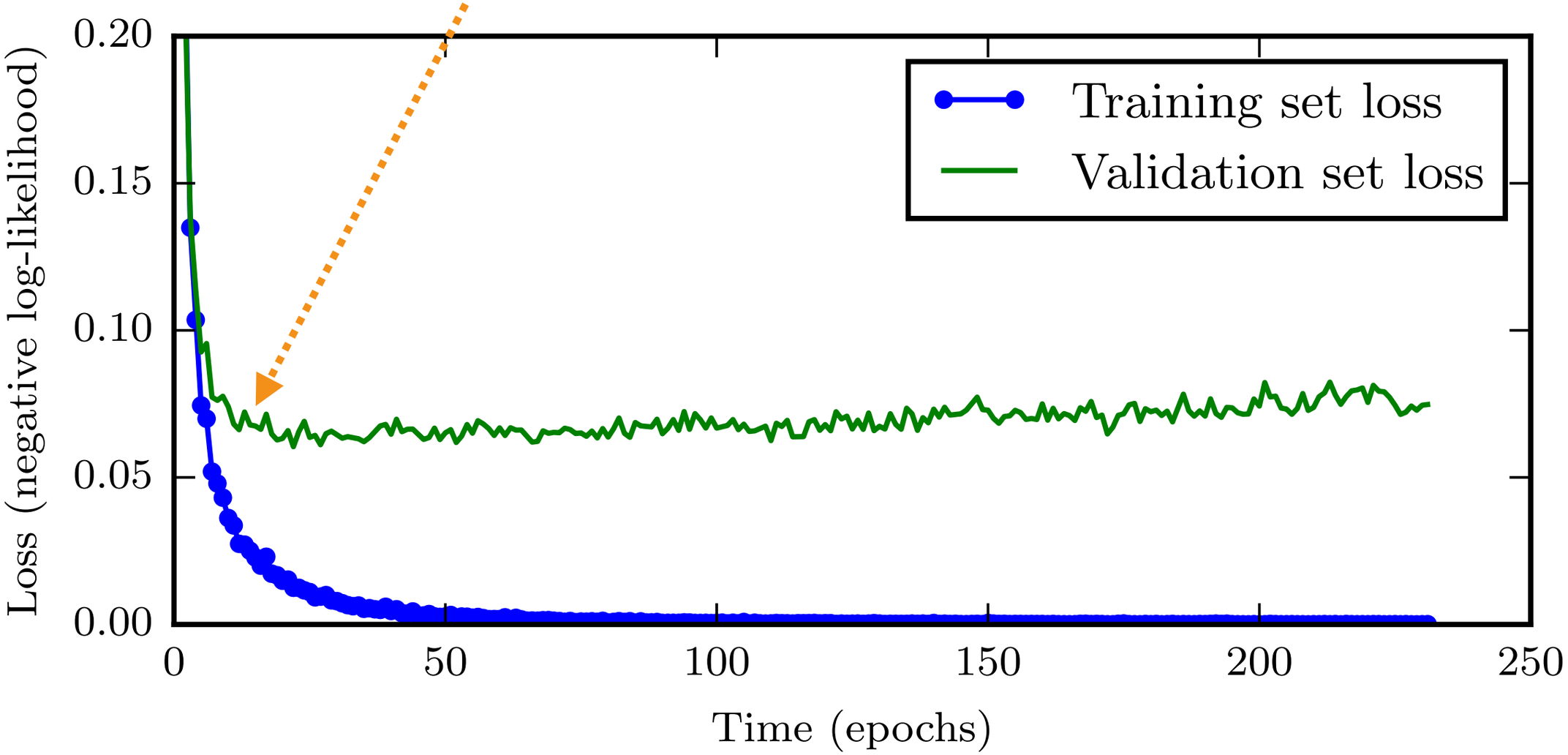
- Neural networks require iterative algorithms for training (typically a gradient descent-type)

- The larger the number of iterations and the lower the training error

- A technique to increase the generalization of the model is to **limit the number of iterations**

Early Stopping

terminate while validation set performance is better



Early Stopping

- Since the validation set is not used for training, after early stopping one can either

  1) retrain the network on all the data (training + validation sets) and then stop after the same number of steps of the early stopping or

  2) continue training the network on all the data (training + validation sets) and then stop when the loss on the validation set is below the loss on the training set (at the early stopping iteration time)