M2177.0043 Introduction to Deep Learning

Lecture 6: Score functions and Loss functions[1]

Hyun Oh Song[1]

[1] Dept. of Computer Science and Engineering, Seoul National University

April 2, 2020

[1] Many slides and figures adapted Justin Johnson

Last time

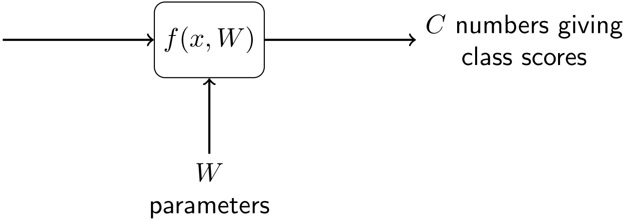
- Subgradient
- Online method

Outline

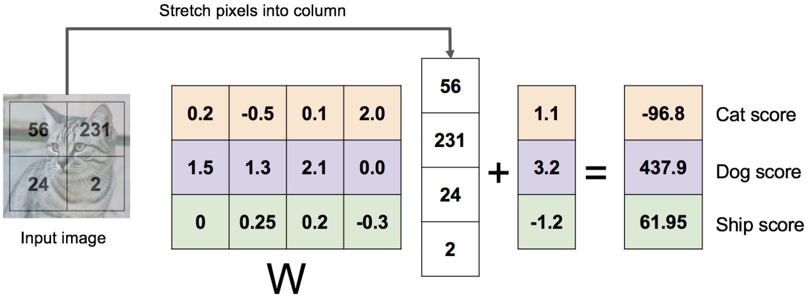
Example score function for C-way classification

$$\underset{C\times 1}{f(x,W)} = \underset{C\times 3072}{W} \overset{3072\times 1}{x} + \underset{C\times 1}{b}$$

Array of 32x32x3
numbers



Example score function for 3-way classification



Interpreting a learned linear classifier

$$f(x,W) = Wx + b$$

Example trained weights of a linear classifier trained on CIFAR-10:

Example scores for 3-way classification

airplane	-3.45	-0.51	3.42
automobile	-8.87	**6.04**	4.64
bird	0.09	5.31	2.65
cat	**2.9**	-4.22	5.1
deer	4.48	-4.19	2.64
dog	8.02	3.58	5.55
frog	3.78	4.49	**-4.34**
horse	1.06	-4.37	-1.5
ship	-0.36	-2.09	-4.79
truck	-0.72	-2.93	6.14

How can we tell whether this W parameter is good or bad?

Outline

Score functions

Loss functions

Loss function

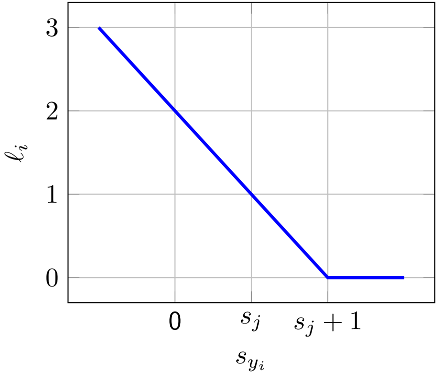
- A loss function is a function that maps its events or values onto a real number intuitively representing some "cost" associated with the event.
- For the classification example, given a dataset of examples $\{(x_i, y_i)\}_{i=1}^{n}$ where x_i is image and y_i is label, loss over the dataset is a sum of loss over examples.

$$\ell(W) = \frac{1}{n}\sum_i \ell_i\left(\underbrace{f(x_i, W)}_{\text{score function}}, \underbrace{y_i}_{\text{label}}\right)$$

Multiclass hinge loss

Given an example (x_i, y_i) where x_i is the input data (*i.e.* image) and y_i is the label, and using the shorthand for the scores vector $s = f(x_i, W)$, the hinge loss has the form

$$\ell_i(W) = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$



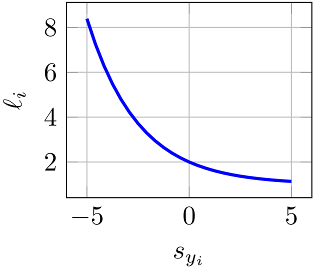
Multinomial logistic loss

View scores as unnormalized log probabilities of the classes.

$$P(Y = k \mid X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

Want to maximize the log likelihood or minimize the negative log likelihood of the correct class.

$$\ell_i(W) = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$



Comparison

$$\ell_i(W) = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

$$\ell_i(W) = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

Assume following scores given the ground truth $y_i = 0$. What happens to the loss if you perturb a datapoint a bit?

$$[\mathbf{10}, -2, 3]$$
$$[\mathbf{10}, 9, 9]$$
$$[\mathbf{10}, -100, -100]$$

Uniqueness

Suppose that we found a W such that the loss is zero. Is this W unique?

$$
\begin{aligned}
f(x, W) &= Wx \\
\ell(W) &= \frac{1}{n}\sum_{i=1}^{n}\sum_{j\neq y_i}\max(0, f(x_i, W)_j - f(x_i, W)_{y_i} + 1)
\end{aligned}
$$

- Multiples of W also satifies *i.e.* $\ell(2W) = 0$

Regularization

$$\ell(W) = \underbrace{\frac{1}{n}\sum_{i=1}^{n} \ell_i(W)}_{\substack{\textbf{Data loss}\text{: model predictions} \\ \text{should match the training data}}}$$

Regularization

$$\ell(W) = \underbrace{\frac{1}{n}\sum_{i=1}^{n} \ell_i(W)}_{\substack{\textbf{Data loss}\text{: model predictions} \\ \text{should match the training data}}} + \underbrace{\lambda R(W)}_{\substack{\textbf{Regularization}\text{: Model should be} \\ \text{simple. so it works on test data}}}$$

Regularization

$$\ell(W) = \underbrace{\frac{1}{n}\sum_{i=1}^{n} \ell_i(W)}_{\substack{\textbf{Data loss}\text{: model predictions} \\ \text{should match the training data}}} + \underbrace{\lambda R(W)}_{\substack{\textbf{Regularization}\text{: Model should be} \\ \text{simple. so it works on test data}}}$$

- ℓ_2 regularization, $R(W) = \|W\|_2$ (connection to MAP inference)
- ℓ_1 regularization, $R(W) = \|W\|_1$ (sparse solution - more on this later on Network Pruning lectures)
- Nuclear (trace) norm, $R(W) = \|W\|_* = \mathbf{Tr}(\sqrt{W^\top W}) = \sum_{i=1}^{\min(m,n)} \sigma_i(W)$
- Dropout, Batch normalization, *etc.*

Regularization

- Regularization function quantifies the complexity of the model and penalizes complex models.
- Simpler models can *underfit*. Complex models can *overfit*.
- How do you choose the regularization constant λ?

Regularization as a constraint[2]

- A p-norm regularized problem can be viewed as solving a constrained problem where the p-norm is less than or equal to some value s.

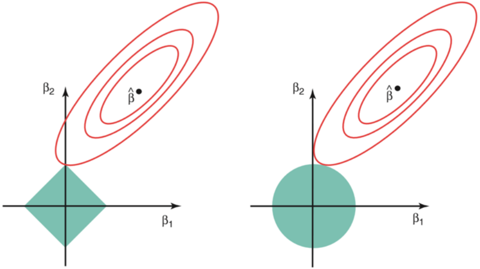


Figure: (Left) ℓ_1 constraint, (Right) ℓ_2 constraint balls. ℓ_1 constraint tends to generate sparser solutions.

[2] Introduction to statistical learning by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani