Softmax Classifier

Softmax loss function

$$L = \sum_k \left[-f_{y_i} + \log \sum_p \exp f_p \right] + \lambda \sum_{i,j} W_{ij}^2$$

k = 1 … num_train number of images

p = 1 … 10 number of classes

$$\frac{\partial L}{\partial W_{ij}} = \sum_k \left[\frac{e^{f_i}}{\sum_p e^{f_p}} - \delta_{j=y_k} \right] (x_k)_i + 2\lambda W_{ij}$$

Numerical stability

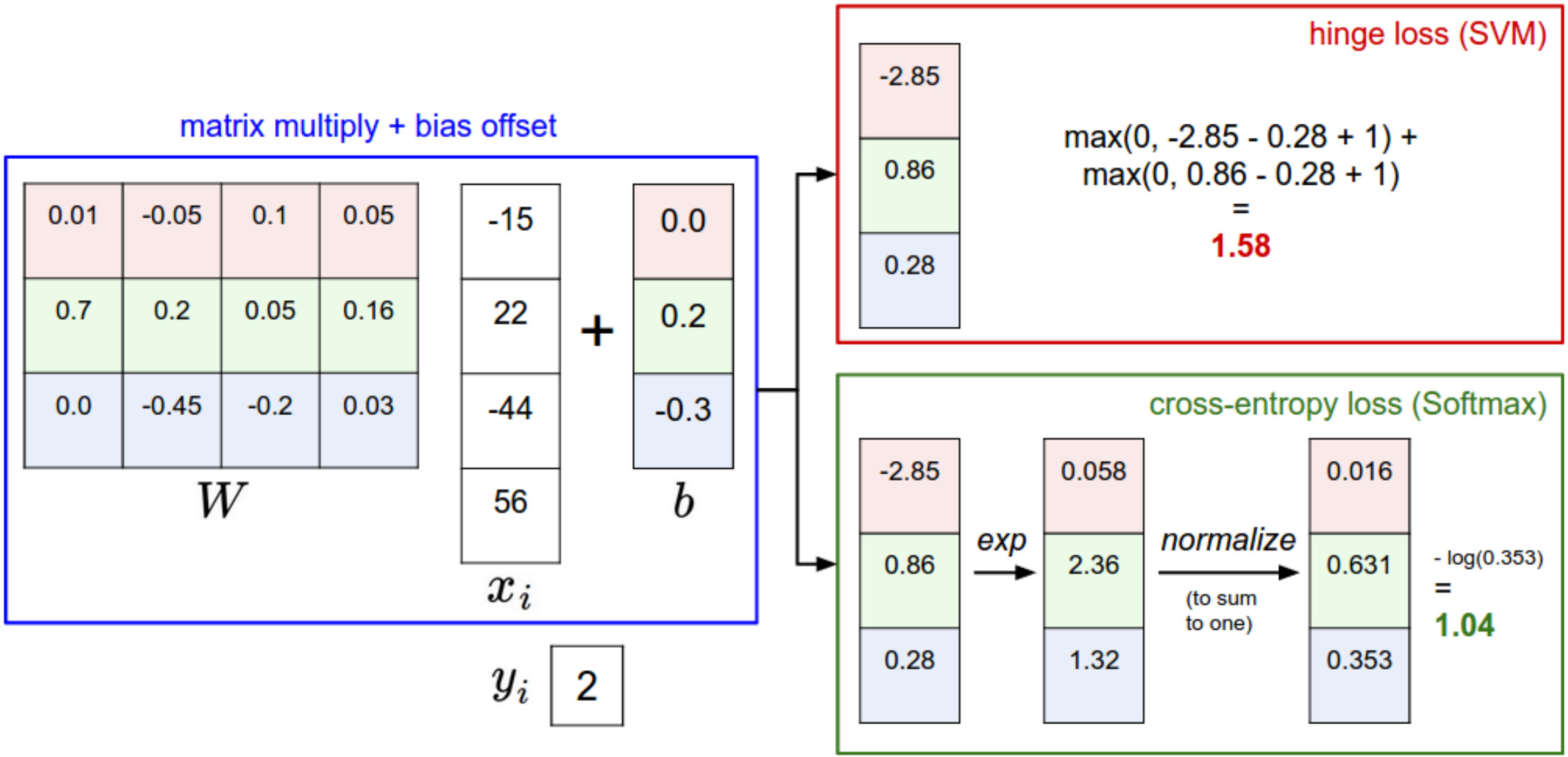
If $f_i = 5$, then $e^{f_i} \approx 148 \approx 10^2$

If $f_i = 7$, then $e^{f_i} \approx 1097 \approx 10^3$

$$\frac{e^{f_i}}{\sum_p e^{f_p}} = \frac{C e^{f_i}}{\sum_p C e^{f_p}} = \frac{e^{f_i + \log C}}{\sum_p e^{f_p + \log C}}$$

Choose $\log C = -\max(f_j)$

then $f_p + \log C \leq 0 \Rightarrow e^{f_i + \log C} \leq 1$


matrix multiply + bias offset
0.01
-0.05
0.1
0.05
0.7
0.2
0.05
0.16
0.0
-0.45
-0.2
0.03
W
-15
22
-44
56
x_i
+
0.0
0.2
-0.3
b
y_i
2
hinge loss (SVM)
-2.85
0.86
0.28
max(0, -2.85 - 0.28 + 1) +
max(0, 0.86 - 0.28 + 1)
=
1.58
cross-entropy loss (Softmax)
-2.85
0.86
0.28
exp
0.058
2.36
1.32
normalize
(to sum
to one)
0.016
0.631
0.353
- log(0.353)
=
1.04