

CU
University of Colorado
Boulder

Deep Learning Applications for Computer Vision

Lecture 13: The Image Classification Pipeline



University of Colorado **Boulder**

Neural Networks for Image Classification

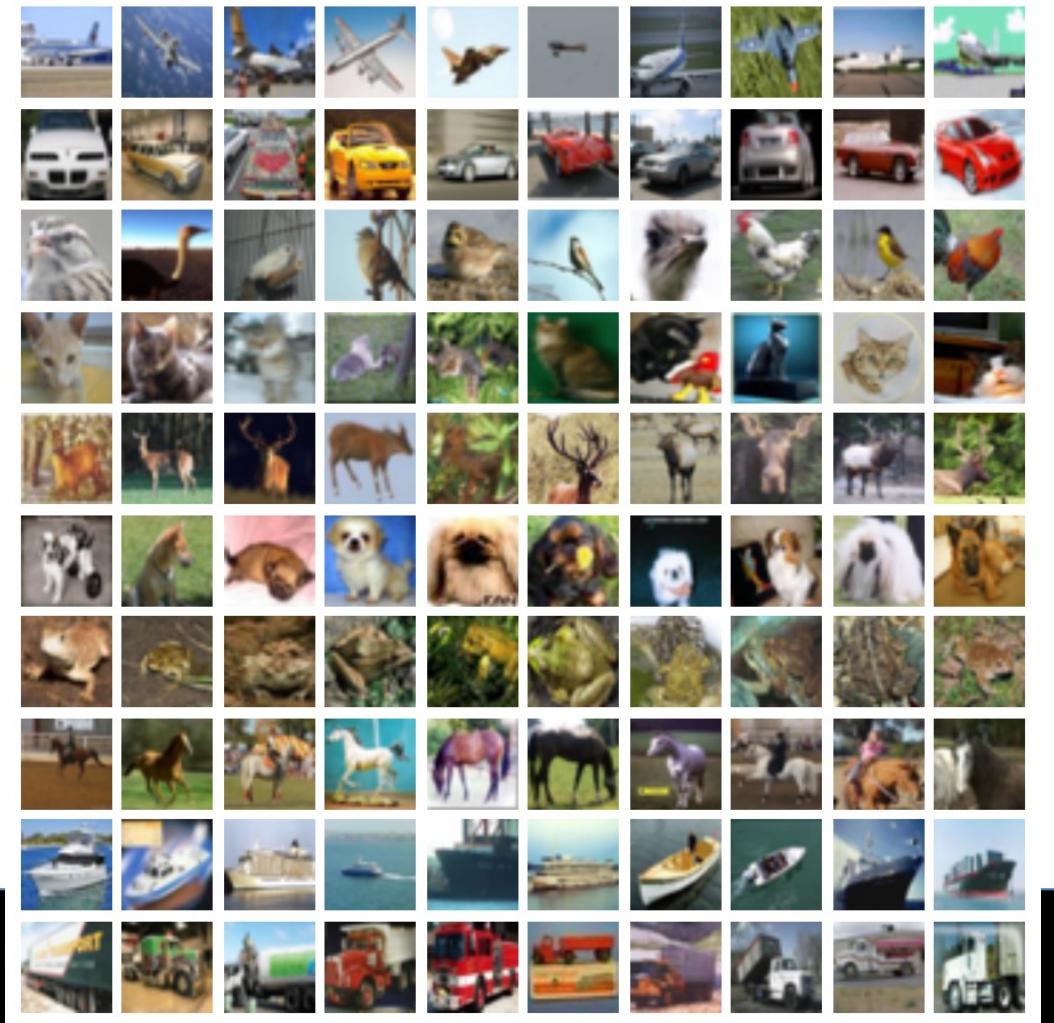
Last time: How do we train a neural network?

Today: Krizhevski, A: “Learning multiple layers of features from tiny images”, 2009

Image Classification with
the CIFAR-10 dataset

(<https://www.cs.toronto.edu/~kriz/cifar.html>)

[C - FAR]



University of Colorado **Boulder**

Neural Networks for Image Classification

Today:

- Example: Image Classification with 10 categories
- CIFAR-10 dataset: “airplane, automobile (but not truck or pickup truck), bird, cat, deer, dog, frog, horse, ship, and truck (but not pickup truck)”
 - 50 000 training images (5000 per class)
 - 10 000 test images
 - image size: 32 x 32 pixels

60k

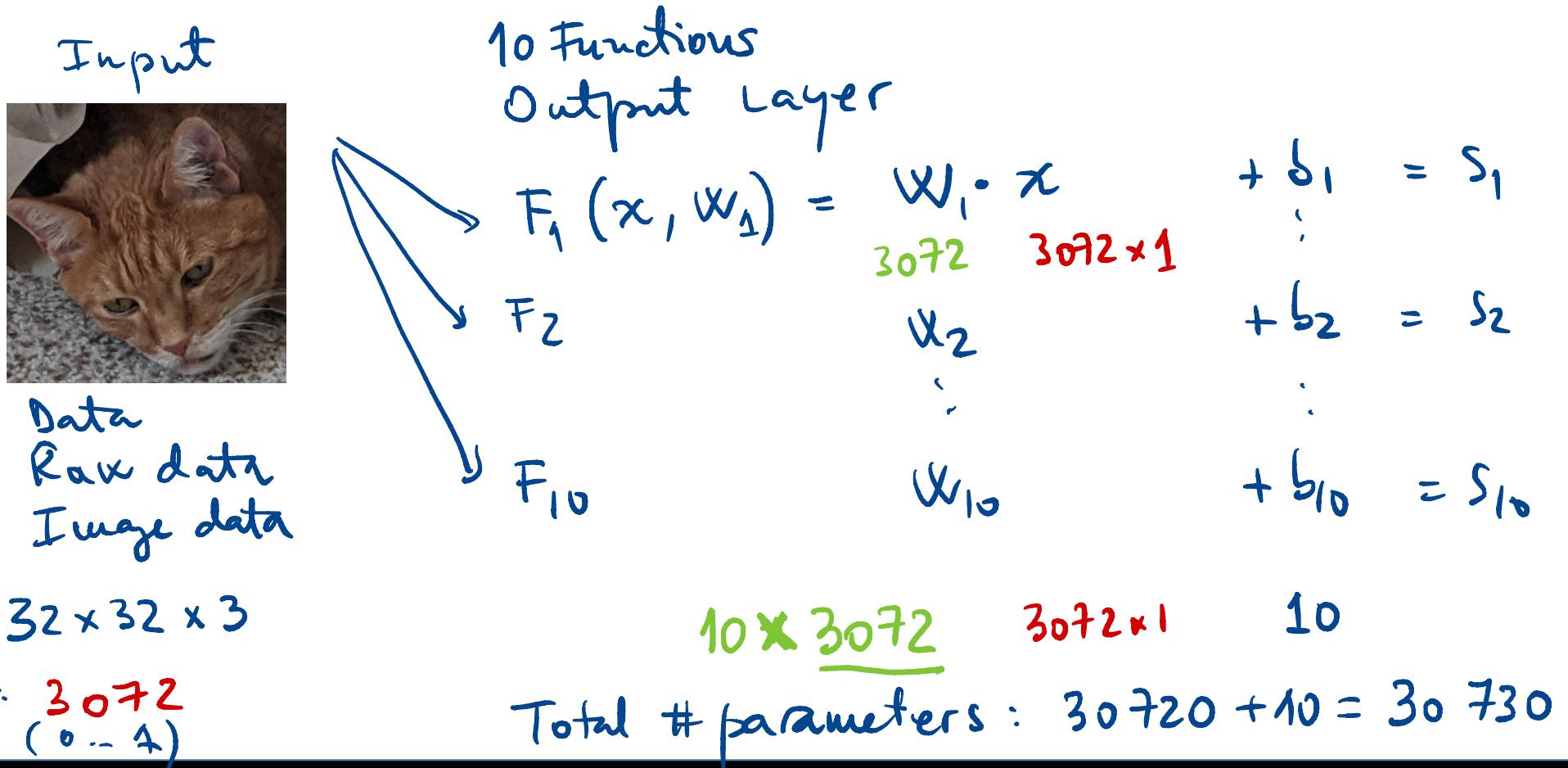
Krizhevski, A, “*Learning multiple layers of features from tiny images*”, 2009



University of Colorado **Boulder**

Network parameters

- Assume only one layer
- Output layer: 10 neurons = 10 labels



University of Colorado **Boulder**

Even Smaller Example

- Image w/ 4 pixel-values
- Classifier w/ 3 labels: frog, dog, cat



$$\begin{array}{c} \text{1) } \begin{matrix} w_1 & w_2 & w_3 & w_4 \\ 0.2 & 0.3 & -0.7 & 0.2 \\ -0.2 & 0.12 & -0.3 & -0.2 \\ 1.2 & 0.3 & 0.01 & -0.9 \end{matrix} * \begin{matrix} x_i \\ 0 \\ 255 \\ 125 \\ 67 \end{matrix} + b = \begin{matrix} s \\ 5.6 \\ -11.6 \\ 9.45 \end{matrix} \\ \text{2) } \text{3) } \end{array}$$



University of Colorado **Boulder**

What happens with the weighted sum?

Cost function, error function, objective function

- Loss function: how happy are we with the scores?

$$P(Y=k | X=x_i) = \frac{e^{s_k}}{\sum e^{s_k}}$$

known label assigned

- Example: Softmax classifier

- takes a vector of arbitrary scores and turns them into values from 0 to 1, which sum to 1
- uses cross-entropy loss: minimize the negative log likelihood of the correct class choice

$$L_i = -\log \left(P(Y=k | X=x_i) \right)$$

Sum to 1
Between 0 and 1

$$s = f(x, w)$$

s_1	5.6
s_2	-11.6
s_3	9.45

$$e^{s_k}$$

0.02
7.06e-10
0.979

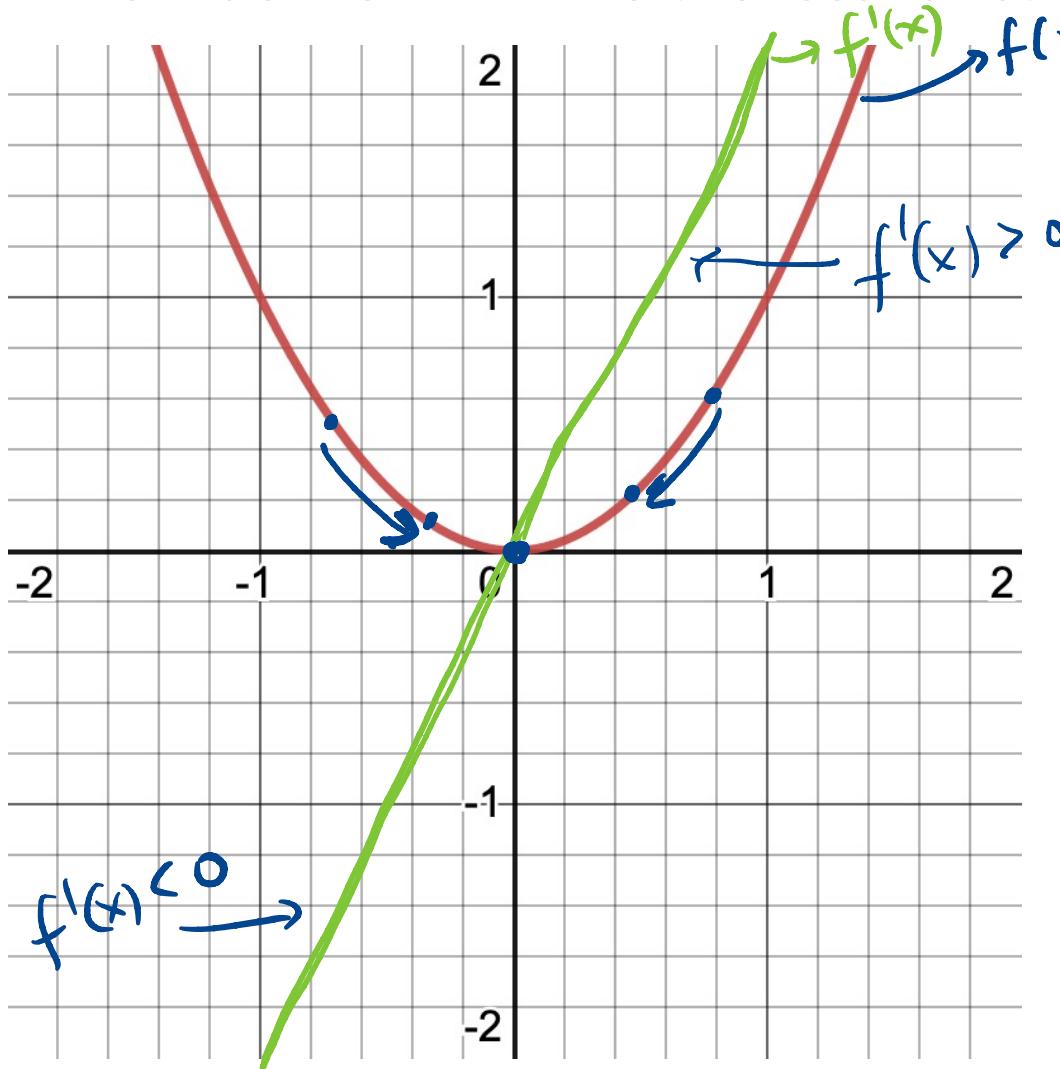
$$\sum e^{s_k}$$

$$-\log(0.979) = \\ = 0.0009$$



Backpropagation and Gradient Descent

- How do we minimize the loss function?



$$f(x) = x^2$$
$$f'(x) = \frac{dy}{dx} = 2x$$

$$f(x + \varepsilon) \approx f(x) + \boxed{\varepsilon * f'(x)} \\ < 0$$

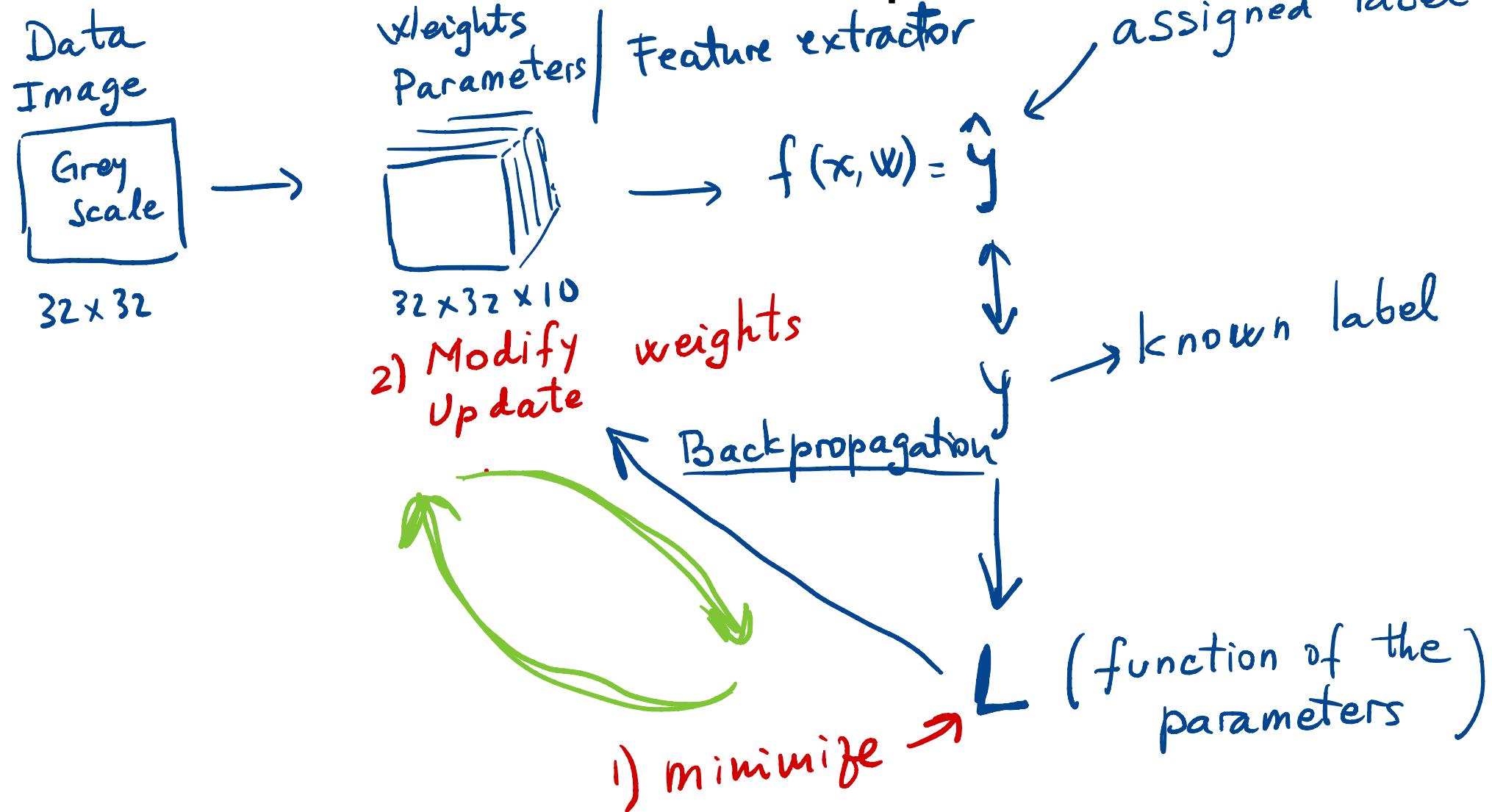
→ if $f'(x) < 0 \Rightarrow \varepsilon > 0$

→ if $f'(x) > 0 \Rightarrow \varepsilon < 0$

Keep moving x in small
steps with the opposite
sign of $f'(x)$ = Gradient
Descent



Classification Pipeline



University of Colorado **Boulder**

System performance

- Time to train
- Overall accuracy
- Best/worst class accuracy



University of Colorado **Boulder**