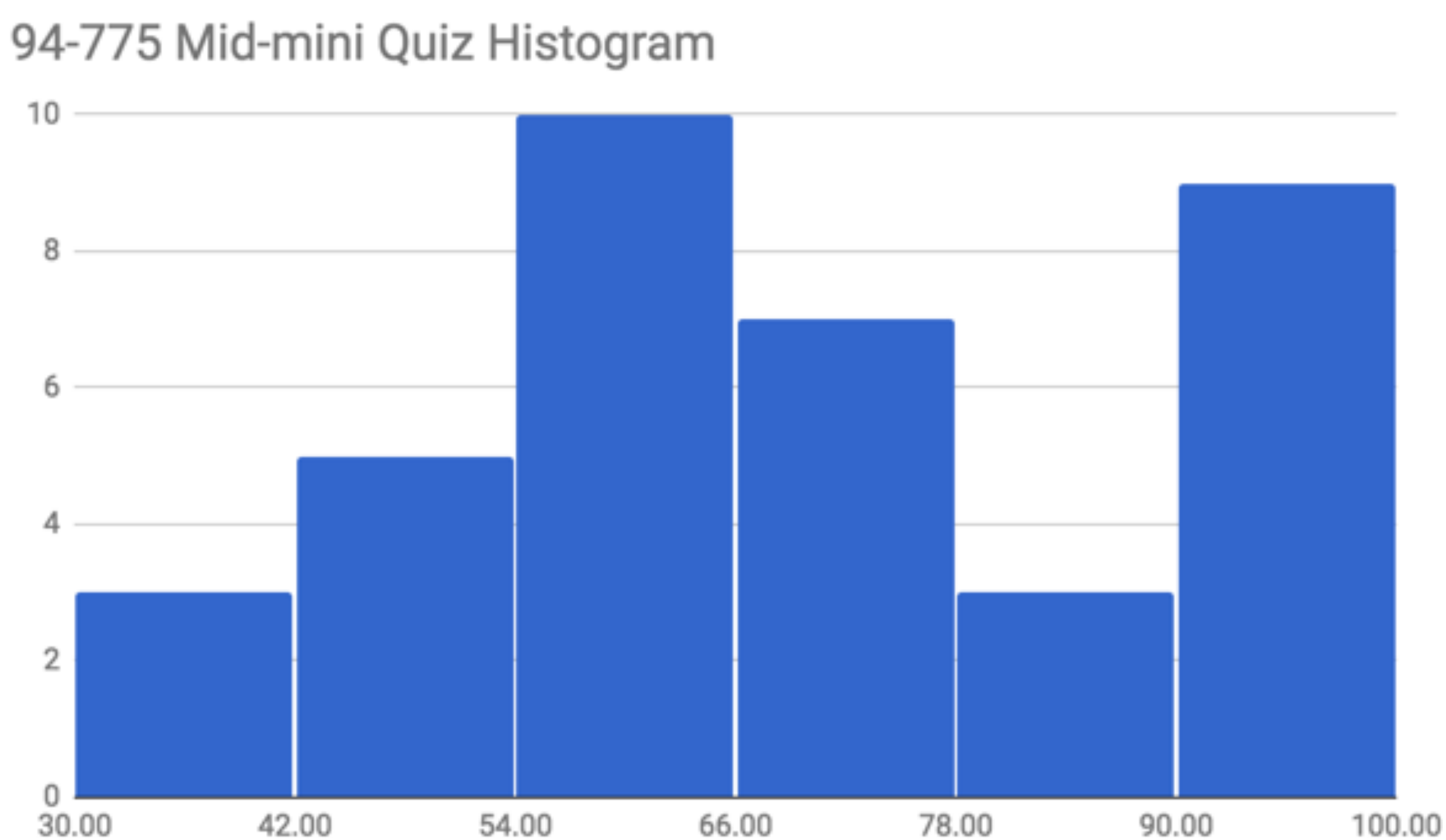
Carnegie Mellon University
Heinz College

94-775 Last Lecture: Wrap-up of Deep Learning and 94-775

nearly all slides by George Chen (CMU)

1 slide by Phillip Isola (OpenAI, UC Berkeley)

Quiz



- Mean: 68.7
- Standard deviation: 19.5
- Max: 99

Some Comments

- This is the first offering of this course!
- I don't know yet what grades will look like
- As this is a pilot course, I plan on leaning more toward the generous side for letter grade assignment
- 84% of students in the class are in the MS PPM program

  There has been a request that MS PPM students be graded on a different curve…

  But all top quiz scores are by MS PPM students!
- Regrettably, grading takes longer than we would like =(
- Next offering of 94-775 has Python as a required pre-req

Final Project Presentation Ordering

Tuesday

1. Arnav Choudhry, James Fasone, Nitin Kumar
2. Rachita Vaidya, Alison Siegel, Eileen Patten, Wei Zhu, Vicky Mei
3. Nattaphat Buddharee, Matthew Jannetti, Angela Wang
4. Hikaru Murase, Nidhi Shree
5. Nicholas Elan, Ben Simmons, Ada Tso, Michael Turner

Thursday

1. Hyung-Gwan Bae, Taimur Farooq, Alvaro Gonzalez, Osama Mansoor, Ben Silliman
2. Quitong Dong, Jun Zhang, Na Su, Wei Huang, Xinlu Yao
3. Anhvinh Doanvo, Wilson Mui, David Pinski, Vinay Srinivasan
4. Jenny Keyt, Natasha Gonzalez, Olga Graves
5. Sicheng Liu, Xi Wang, Jing Zhao

What does analyzing images have to do with policy questions?

Flashback slide: Electrification

Where should we install cost-effective solar panels in developing countries?

Data

- Power distribution data for existing grid infrastructure
- Survey of electricity needs for different populations
- Labor costs
- Raw materials costs (e.g., solar panels, batteries, inverters)
- Satellite images deep nets can be very helpful here!

Related Q: where should a local government extend grid access?

Increasingly easier to get: drone images!

Example: Transportation

Let’s say we’re introducing a new highway route, or a new mode of transportation entirely to get from A to B

How does traffic change on an existing highway from A to B?

Possible data source: fly a drone over a road/highway segment and take images during different times of the day

Unstructured data analysis:

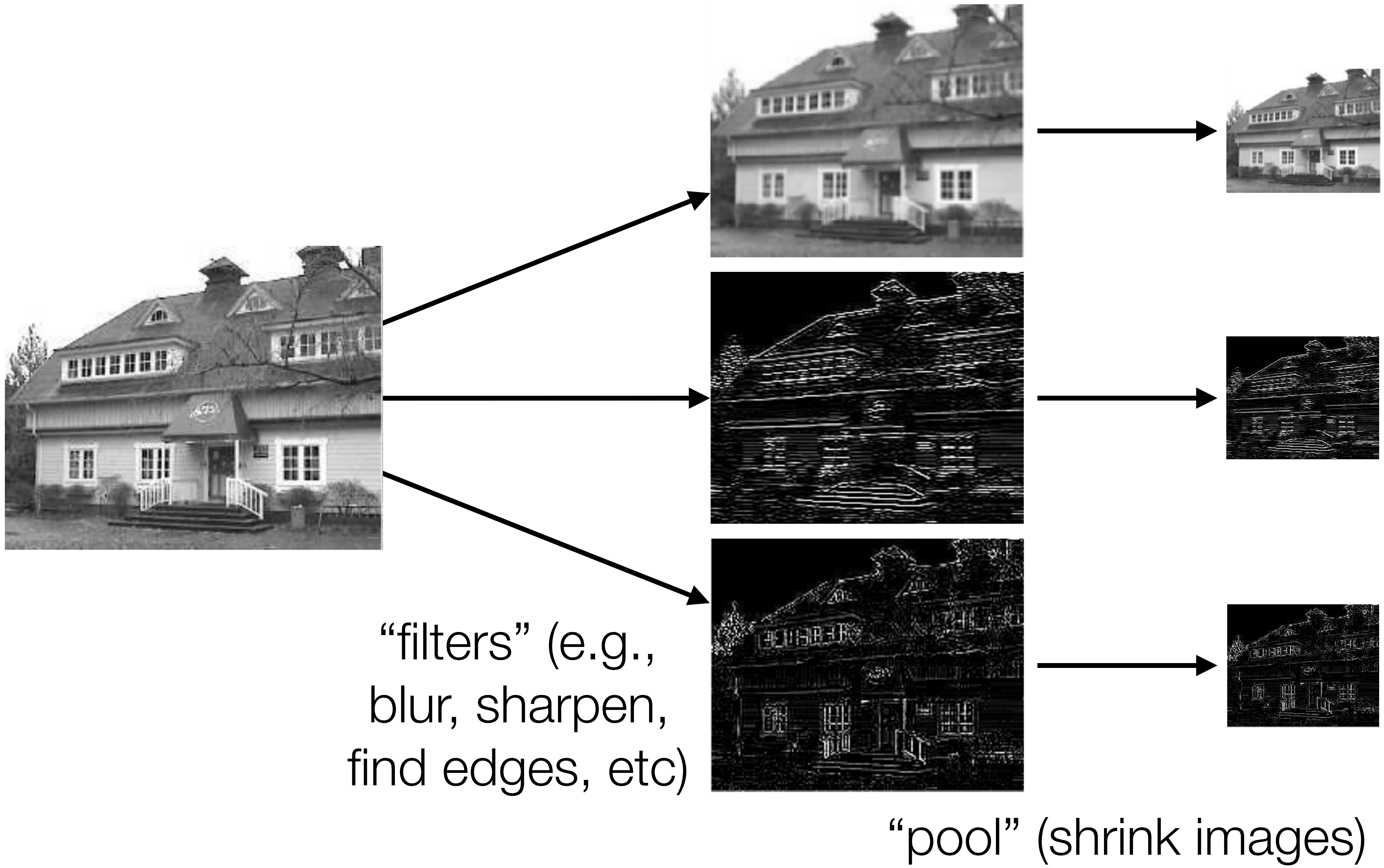
- count cars in images
- distinguish between different types of cars
- come up with throughput estimate

Today

- High-level overview of a bunch of deep learning topics we didn't cover
- (If time) How learning a deep net roughly works
- Course wrap-up

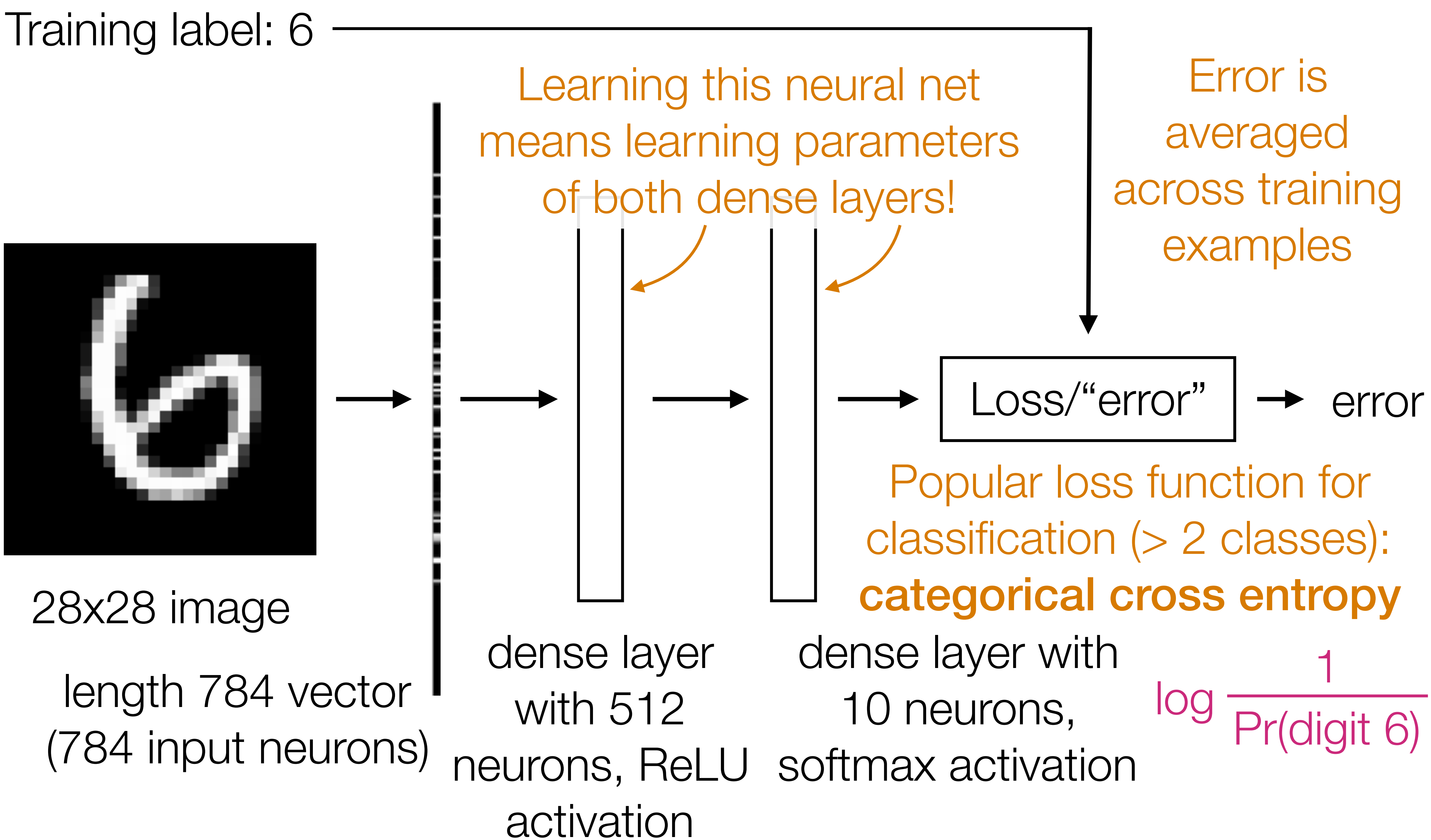
There's a lot more to deep learning that we didn't cover

Image Analysis with CNNs

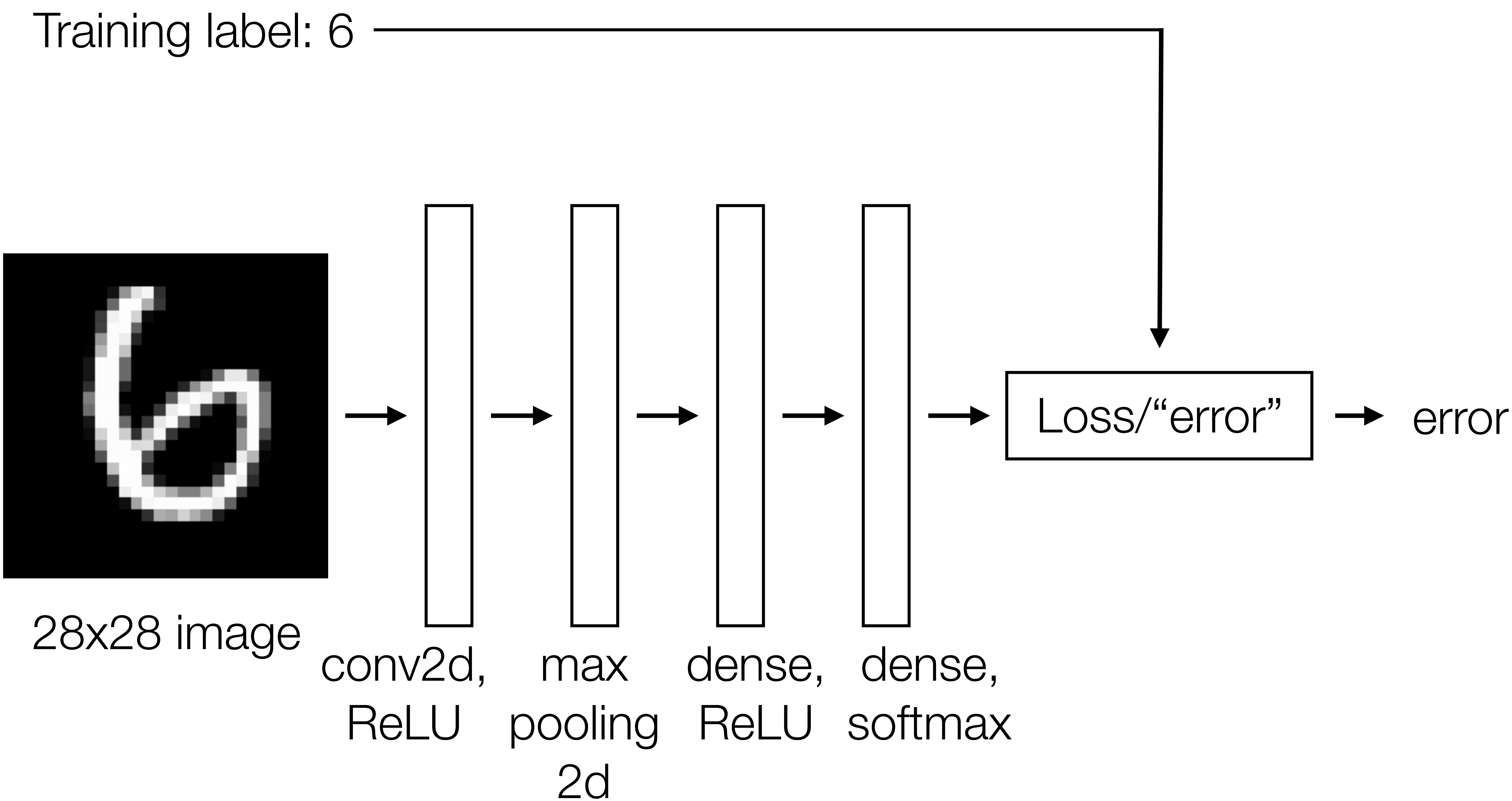


Images from: http://aishack.in/tutorials/image-convolution-examples/

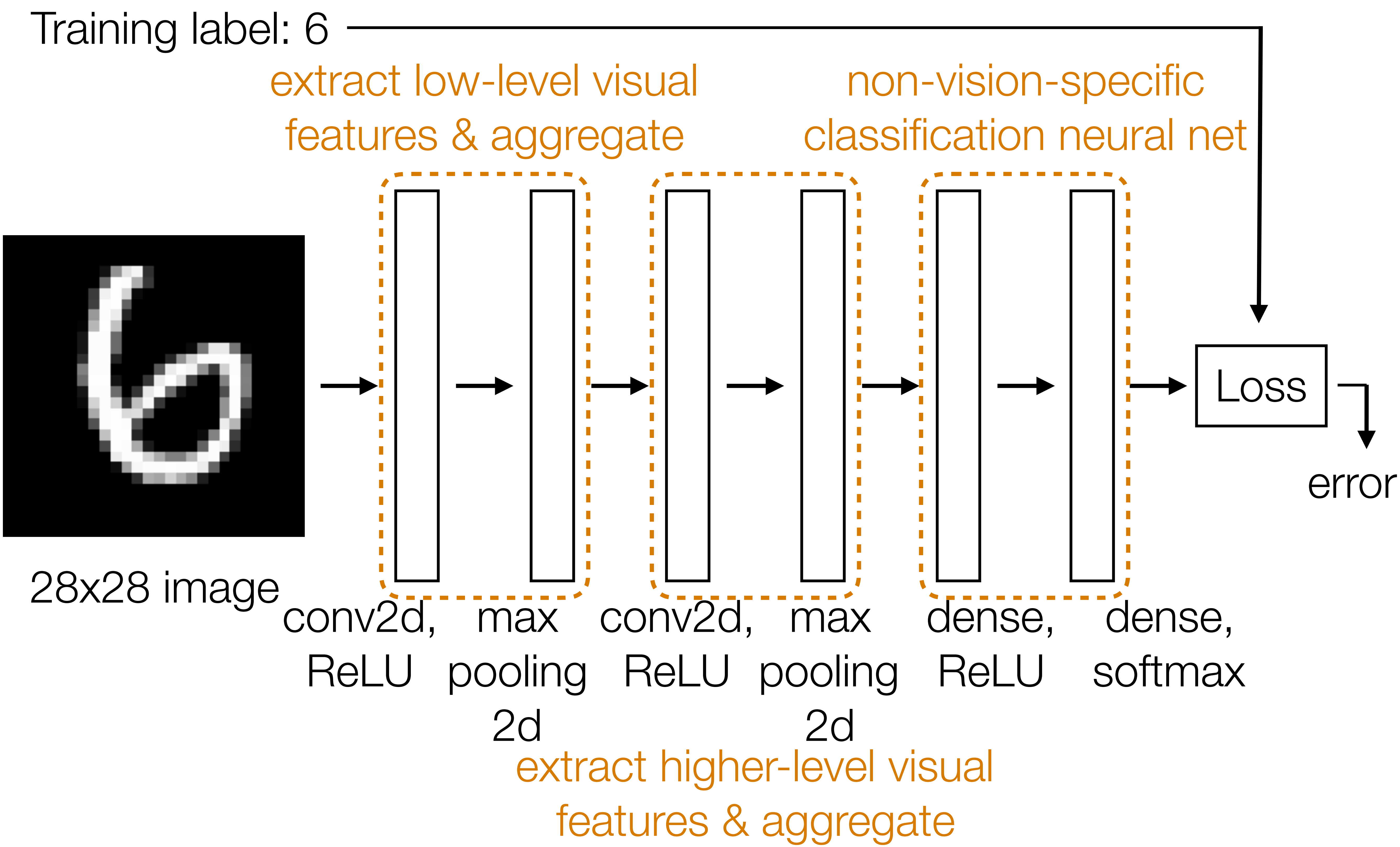
Handwritten Digit Recognition



Handwritten Digit Recognition

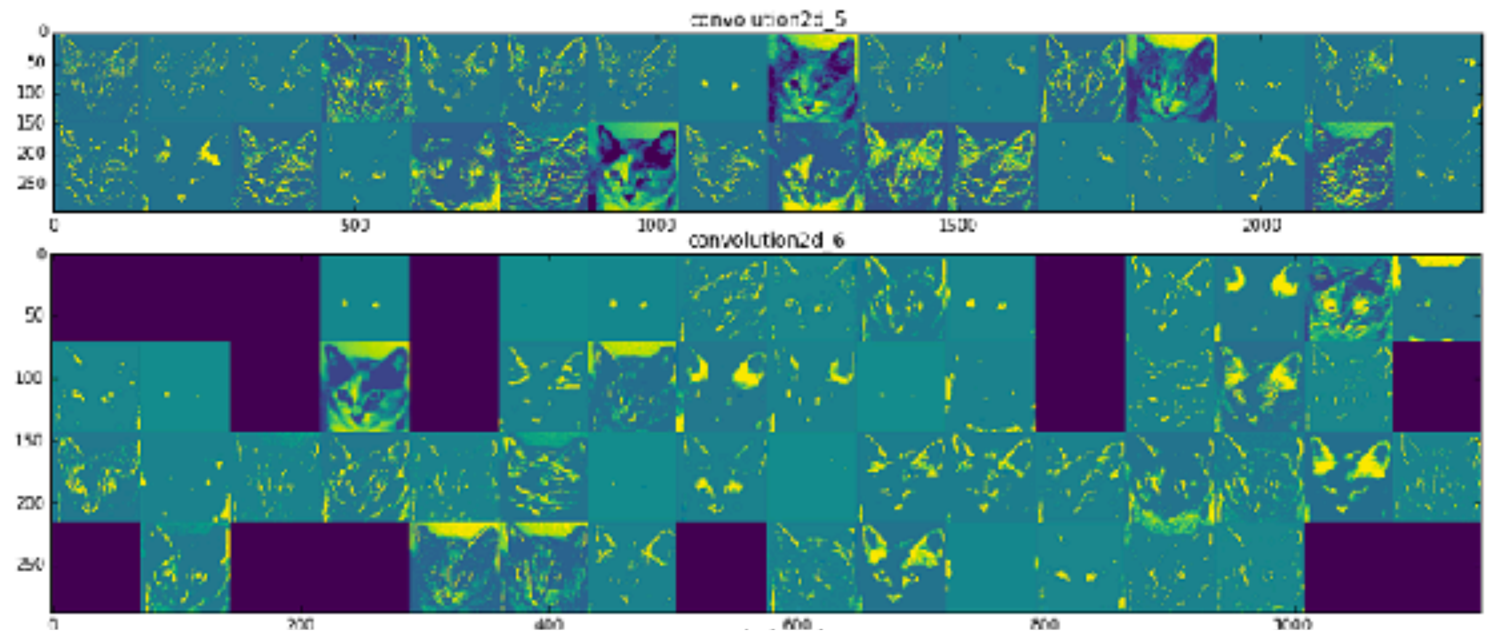


Handwritten Digit Recognition



Visualizing What a CNN Learned

- Plot filter outputs at different layers



- Plot regions that maximally activate an output neuron

Images: Francois Chollet's "Deep Learning with Python" Chapter 5

Example: Wolves vs Huskies

(a) Husky classified as wolf (b) Explanation

Turns out the deep net learned that wolves are wolves because of snow…

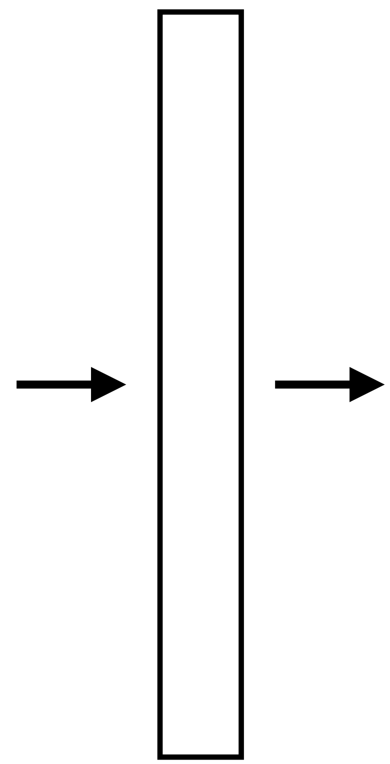
➔ visualization is crucial!

Source: Ribeiro et al. "Why should I trust you? Explaining the predictions of any classifier." KDD 2016.

Time series analysis with Recurrent Neural Networks (RNNs)

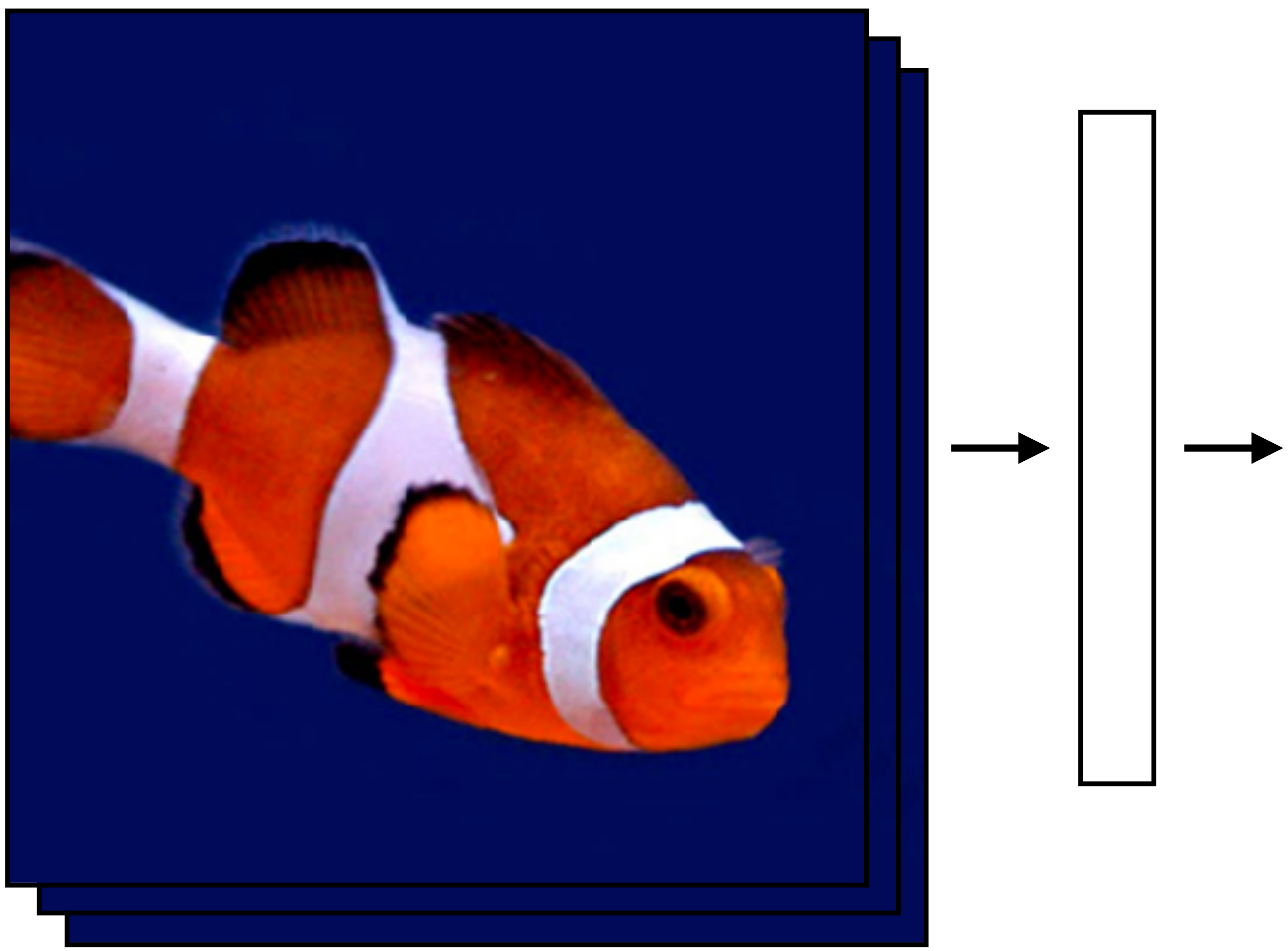
RNNs

What we've seen so far are "feedforward" NNs



RNNs

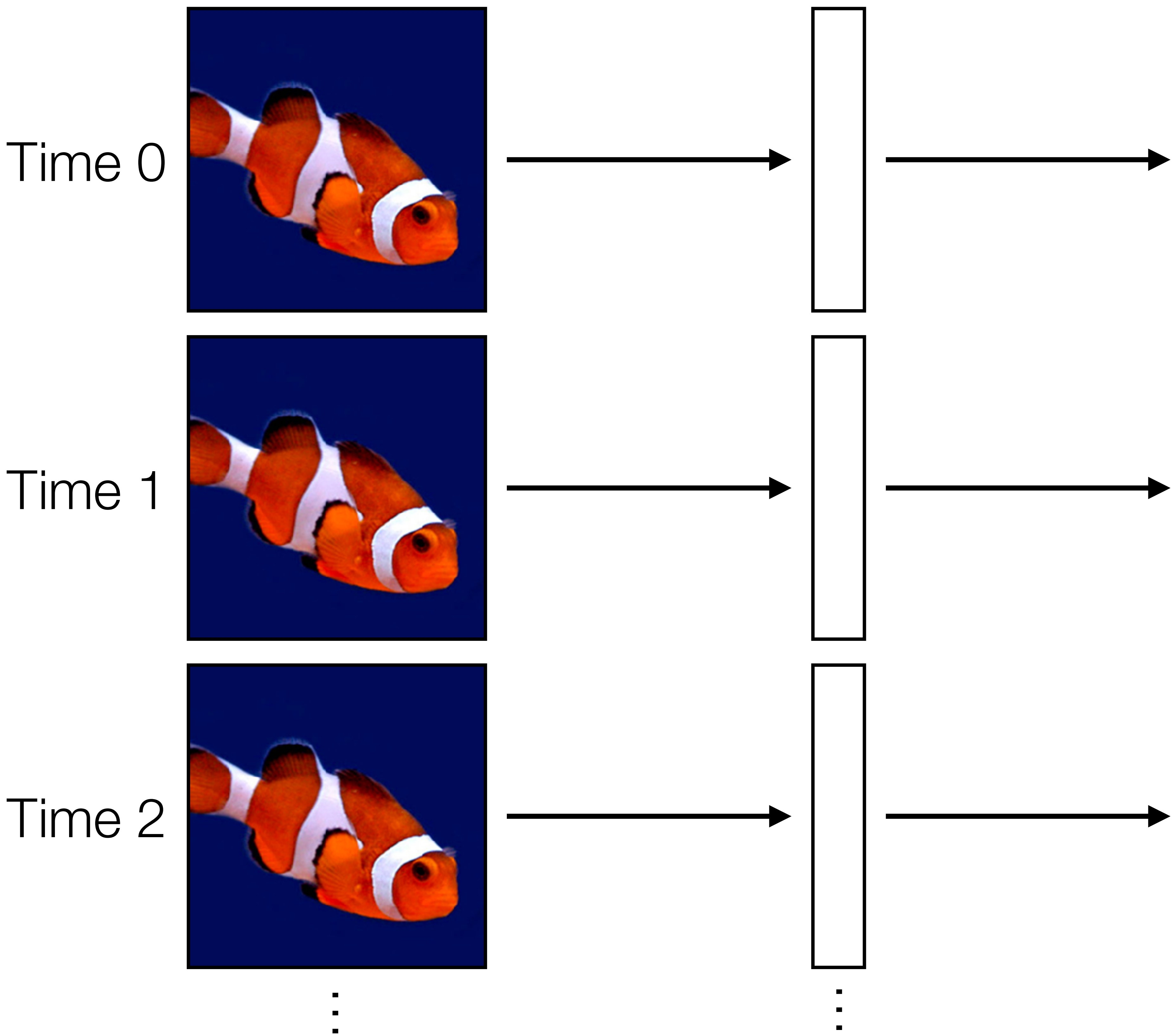
What we've seen so far are "feedforward" NNs



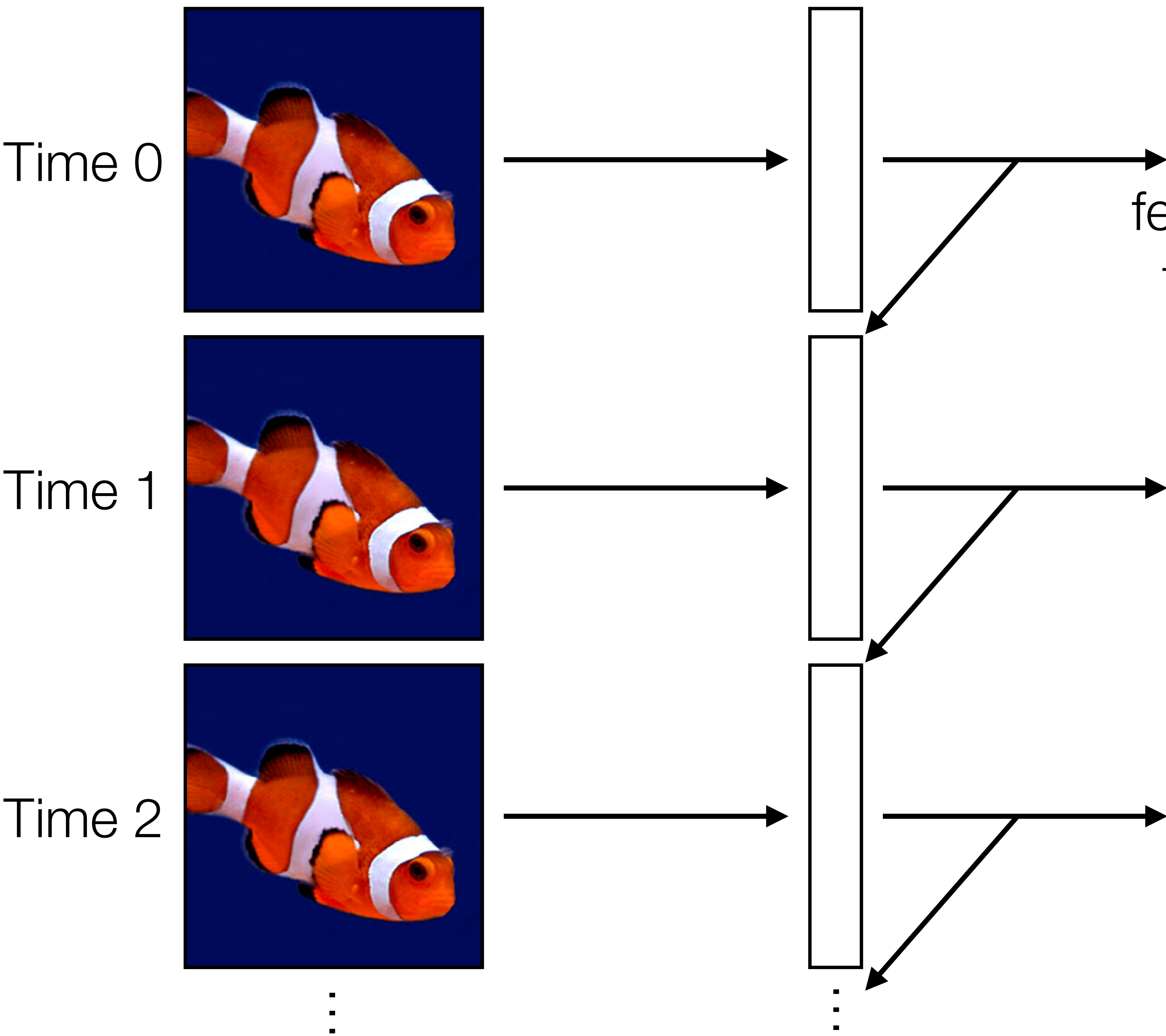
What if we had a video?

RNNs

Feedforward NN's:
treat each video frame
separately



RNNs



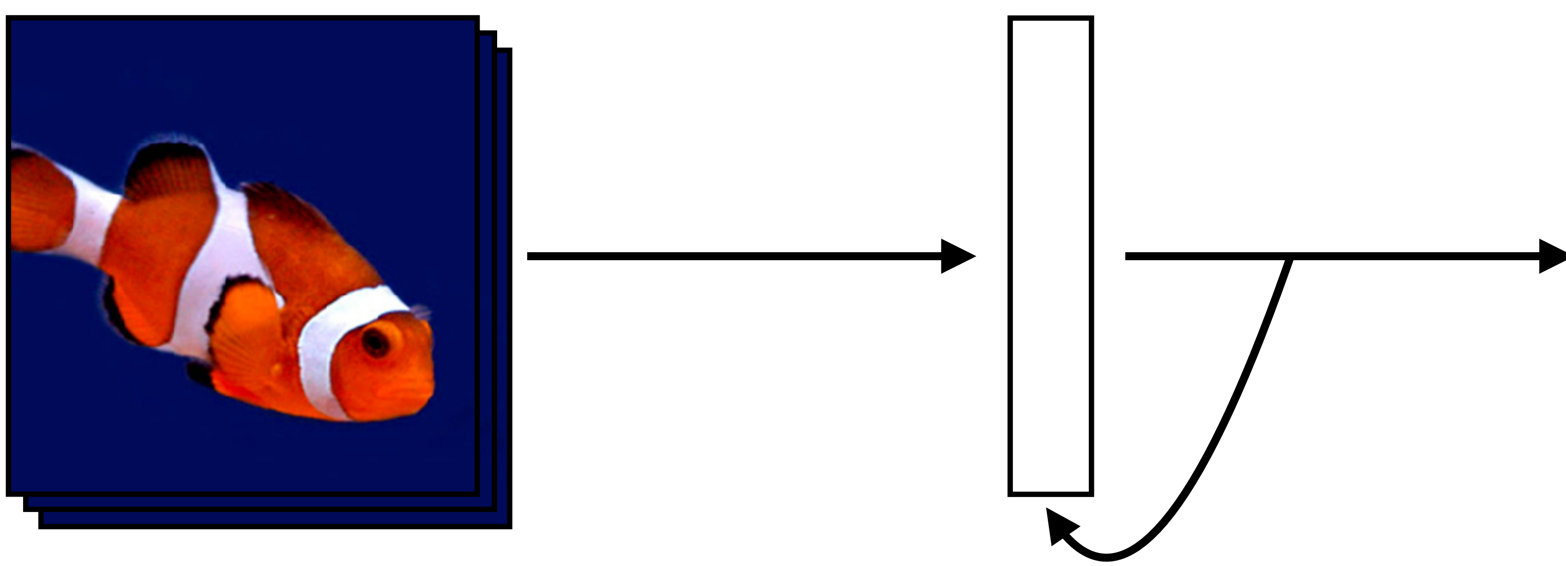
Feedforward NN's:
treat each video frame separately

RNN's:
feed output at previous time step as input to RNN layer at current time step

In `keras`, different RNN options:
`SimpleRNN`, `LSTM`, `GRU`

RNNs

readily chains together with
other neural net layers



Time series

LSTM layer

like a dense layer
that has memory

Feedforward NN's:
treat each video frame separately

RNN's:
feed output at previous time step as input to RNN layer at current time step

In `keras`, different RNN options:
`SimpleRNN`, `LSTM`, `GRU`

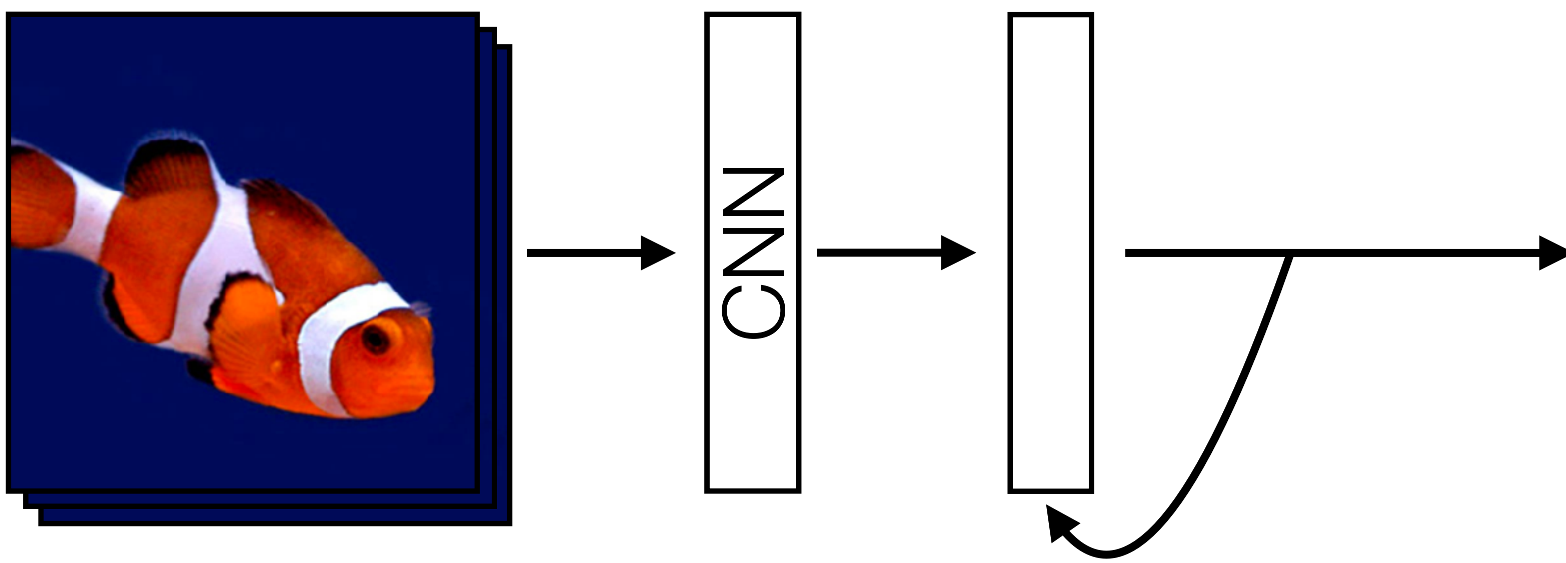
RNNs

Feedforward NN's:
treat each video frame separately

RNN's:
feed output at previous time step as input to RNN layer at current time step

In `keras`, different RNN options:
`SimpleRNN`, `LSTM`, `GRU`

readily chains together with other neural net layers



Time series

LSTM layer

like a dense layer that has memory

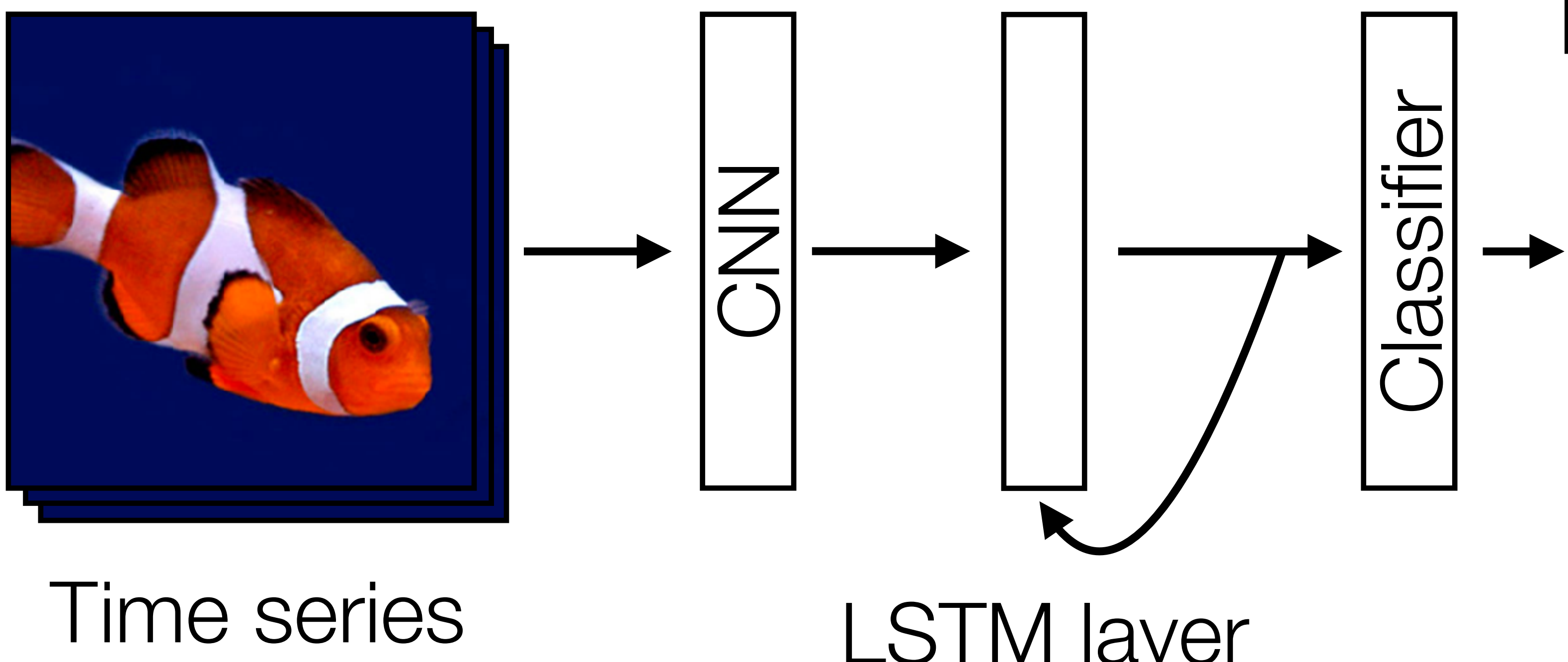
RNNs

Feedforward NN's:
treat each video frame separately

RNN's:
feed output at previous time step as input to RNN layer at current time step

In `keras`, different RNN options: `SimpleRNN`, `LSTM`, `GRU`

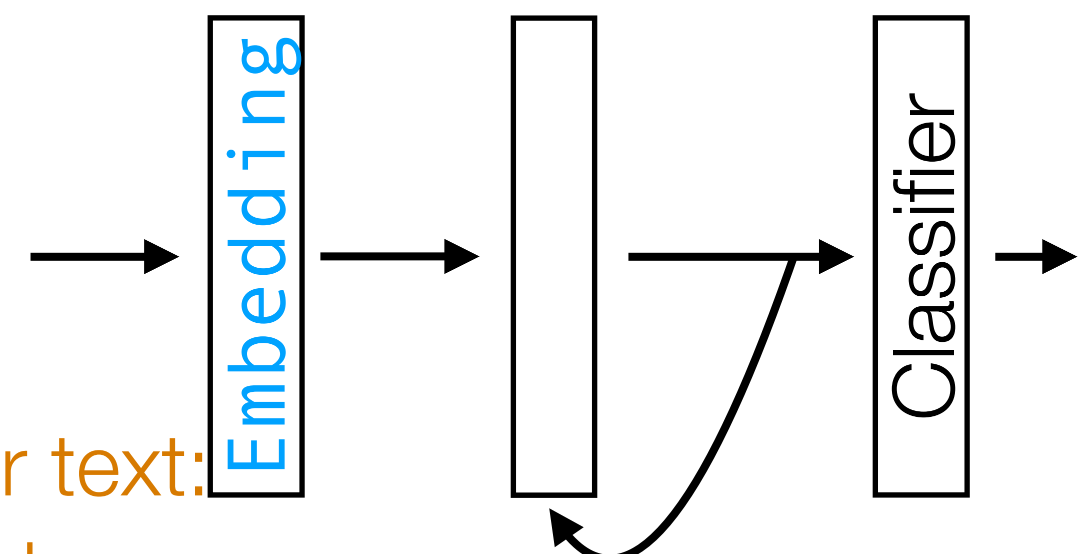
readily chains together with other neural net layers



like a dense layer that has memory

RNNs

Example: Given text (e.g., movie review, Tweet), figure out whether it has positive or negative sentiment (binary classification)



Common first step for text: turn words into vector representations that are semantically meaningful

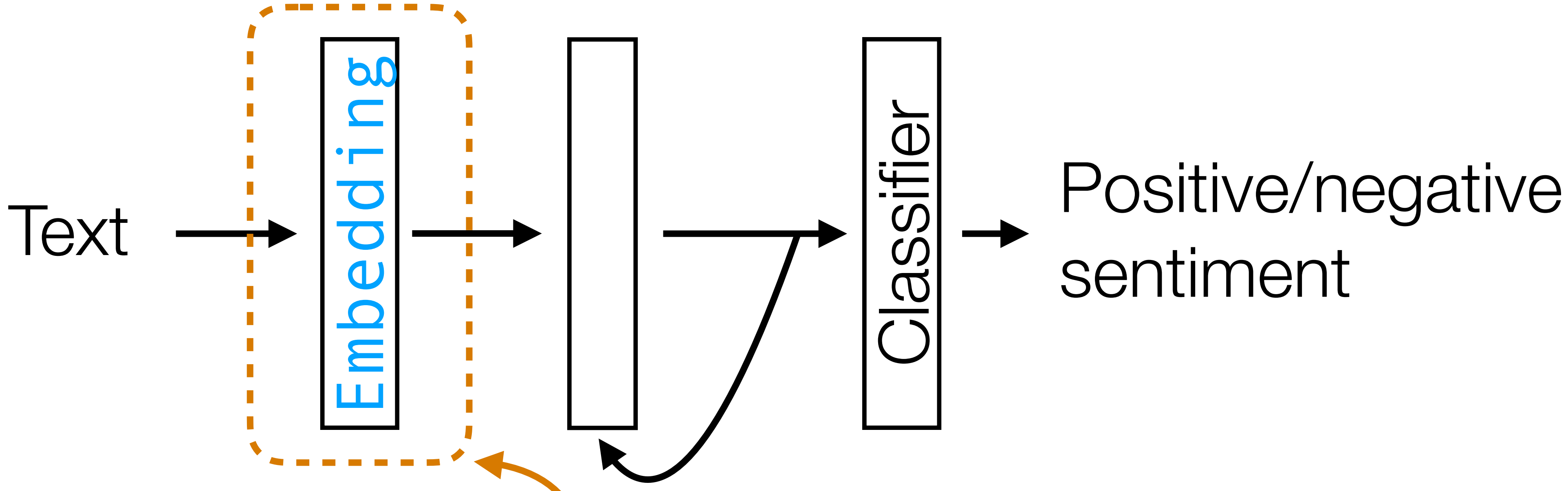
In `keras`, use the `Embedding` layer

Classification with > 2 classes: dense layer, softmax activation

Classification with 2 classes: dense layer with 1 neuron, sigmoid activation

Dealing with Small Datasets

Fine tuning: if there's an existing pre-trained neural net, you could modify it for your problem that has a small dataset



We fix weights here to come from GloVe and disable training for this layer!

GloVe vectors pre-trained on massive dataset (Wikipedia + Gigaword)

Actual dataset you want to do sentiment analysis on can be smaller

Dealing with Small Datasets

Data augmentation: generate perturbed versions of your training data to get larger training dataset

Training image
Training label: cat

Mirrored
Still a cat!

Rotated & translated
Still a cat!

We just turned 1 training example in 3 training examples

Allowable perturbations depend on data (e.g., for handwritten digits, rotating by 180 degrees would be bad: confuse 6's and 9's)

Self-Supervised Learning

Even without labels, we can set up a prediction task!

Example: word embeddings like word2vec, GloVe

The opioid epidemic or opioid crisis is the rapid increase in the use of prescription and non-prescription opioid drugs in the United States and Canada in the 2010s.

Predict context of each word!

Training data point: epidemic

“Training label”: the, opioid, or, opioid

Self-Supervised Learning

Even without labels, we can set up a prediction task!

Example: word embeddings like word2vec, GloVe

The opioid epidemic or opioid crisis is the rapid increase in the use of prescription and non-prescription opioid drugs in the United States and Canada in the 2010s.

Predict context of each word!

Training data point: or

“Training label”: opioid, epidemic, opioid, crisis

Self-Supervised Learning

Even without labels, we can set up a prediction task!

Example: word embeddings like word2vec, GloVe

The opioid epidemic or opioid crisis is the rapid increase in the use of prescription and non-prescription opioid drugs in the United States and Canada in the 2010s.

Predict context of each word!

There are “positive” examples of what context words are for “opioid”

Training data point: opioid

“Training label”: epidemic, or, crisis, is

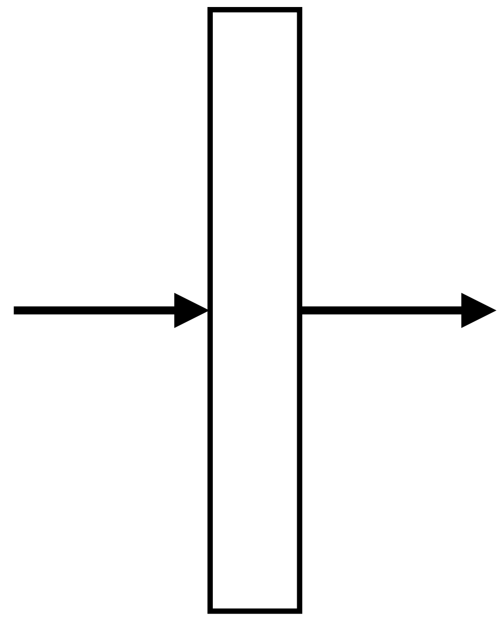
Also provide “negative” examples of words that are *not* likely to be context words (e.g., randomly sample words elsewhere in document)

Self-Supervised Learning

Even without labels, we can set up a prediction task!

Example: word embeddings like word2vec, GloVe

Input word (categorical “one hot” encoding)



Vector saying the probabilities of different words being context words

Dense layer, softmax activation

This actually relates to PMI!

Weight matrix: (# words in vocab) by (# neurons)

Dictionary word *i* has “word embedding” given by row *i* of weight matrix

Self-Supervised Learning

Even without labels, we can set up a prediction task!

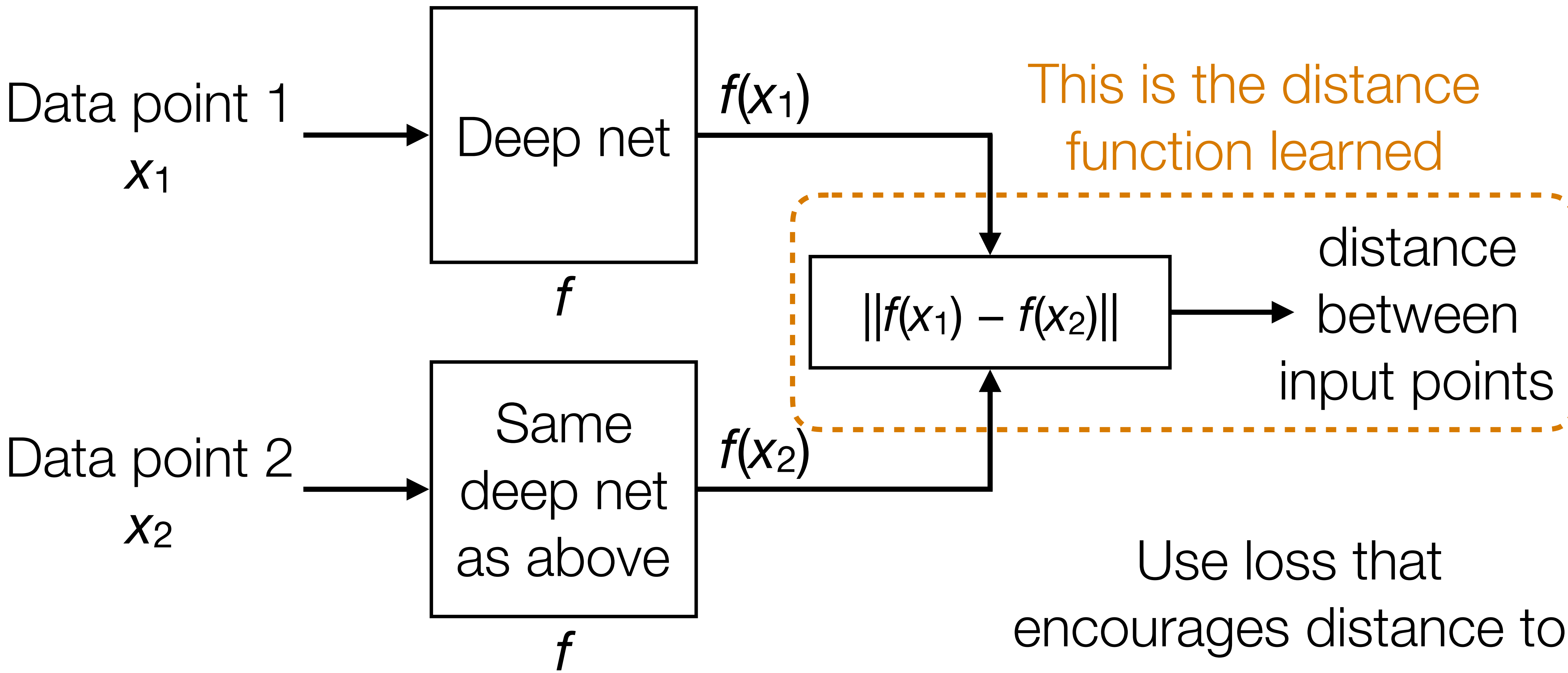
- Key idea: predict part of the training data from other parts of the training data

- No actual training labels required — we are defining what the training labels are just using the unlabeled training data

- This is an *unsupervised* method that sets up a *supervised prediction* task

Learning Distances with Siamese Nets

Using labeled data, we can learn a distance function



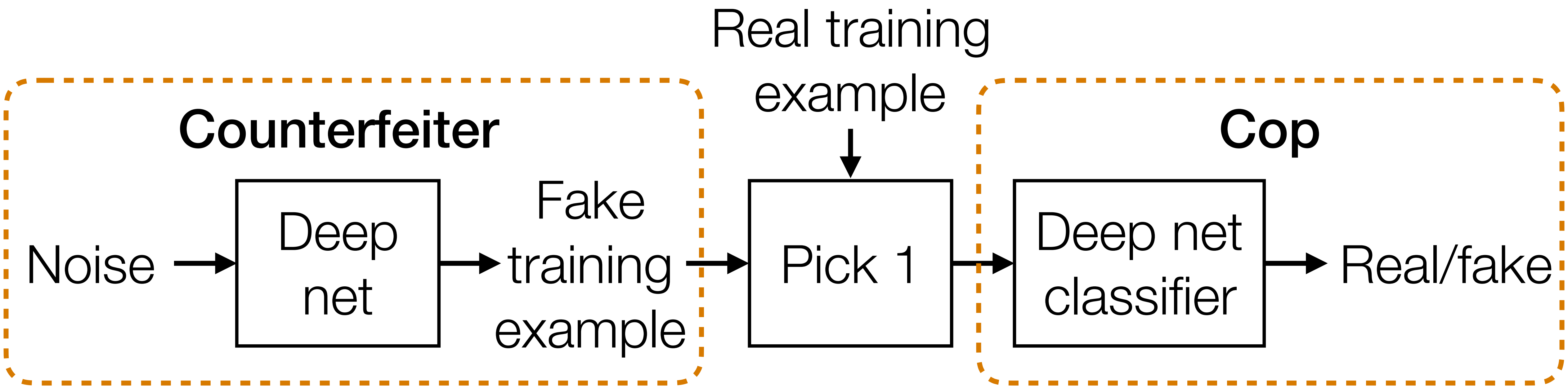
Note: we are learning the function f

Use loss that encourages distance to be small for data points with same label and large otherwise

Generate Fake Data that Look Real

Unsupervised approach: generate data that look like training data

Example: Generative Adversarial Network (GAN)



Counterfeiter tries to get better at tricking the cop

Cop tries to get better at telling which examples are real vs fake

Terminology: counterfeiter is the **generator**, cop is the **discriminator**

Other approaches: variational autoencoders, pixelRNNs/pixelCNNs

Generate Fake Data that Look Real

Fake celebrities generated by NVIDIA using GANs
(Karras et al Oct 27, 2017)

Google DeepMind's WaveNet makes fake audio that sounds like whoever you want using pixelRNNs (Oord et al 2016)

Generate Fake Data that Look Real

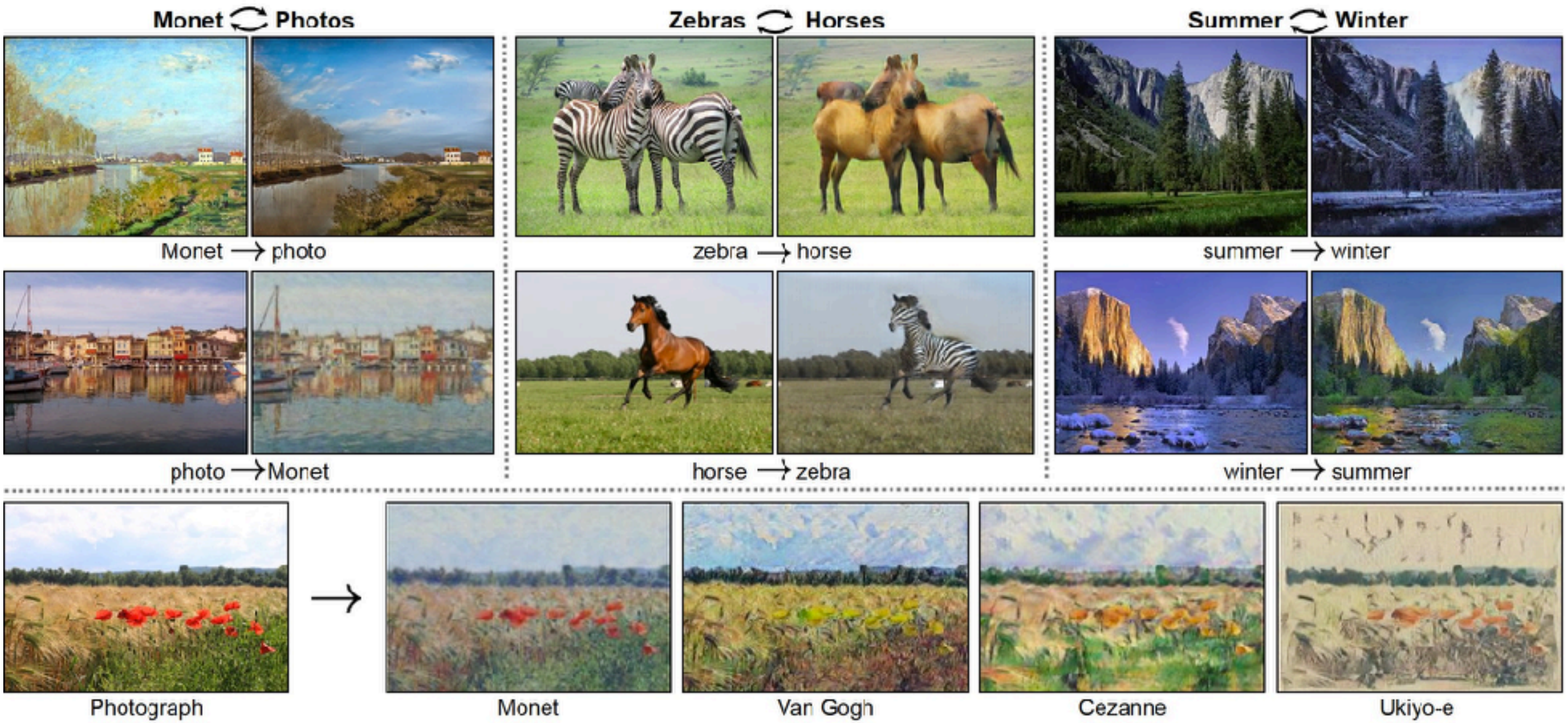
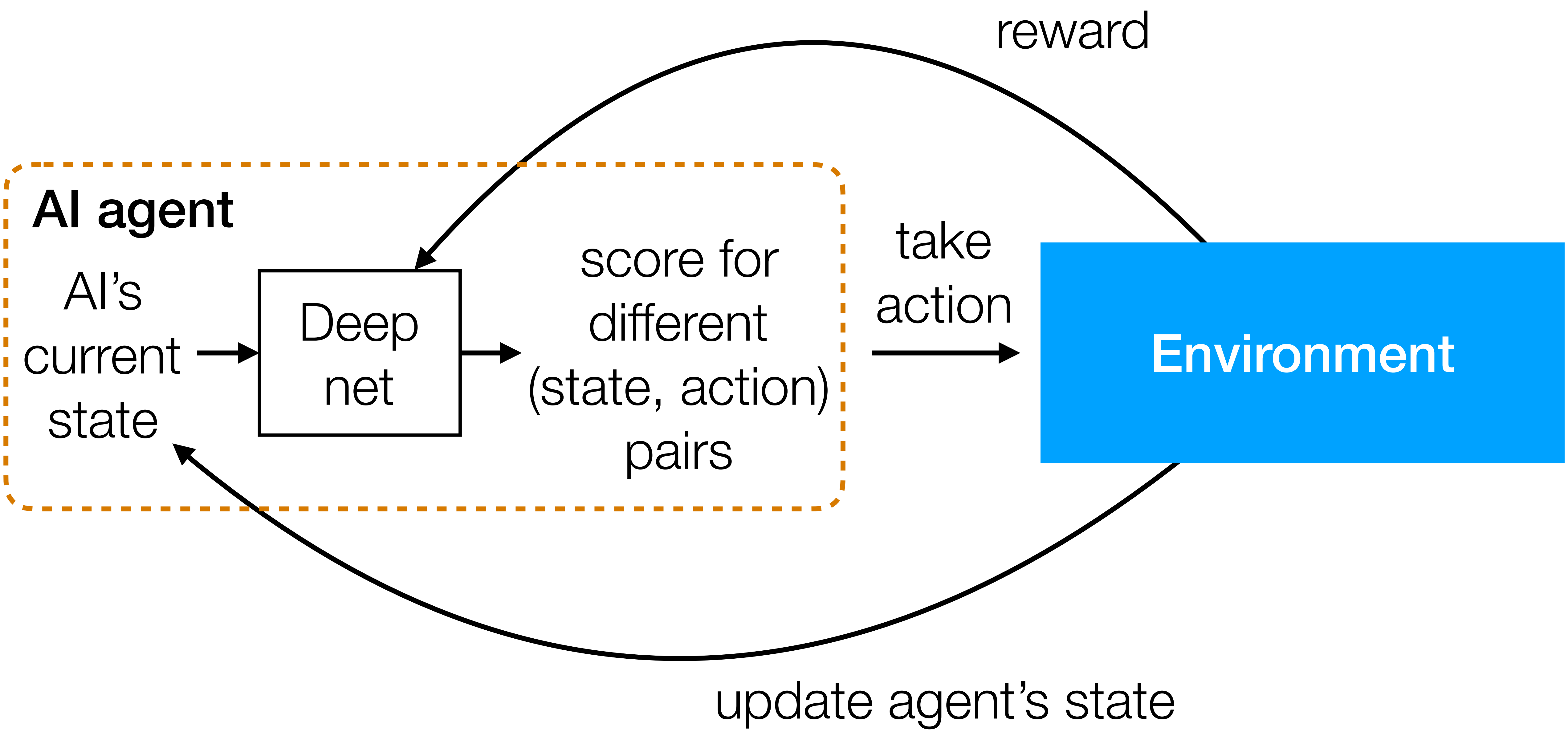


Image-to-image translation results from UC Berkeley using GANs
(Isola et al 2017, Zhu et al 2017)

Deep Reinforcement Learning

The machinery behind AlphaGo and similar systems



Learning a Deep Net

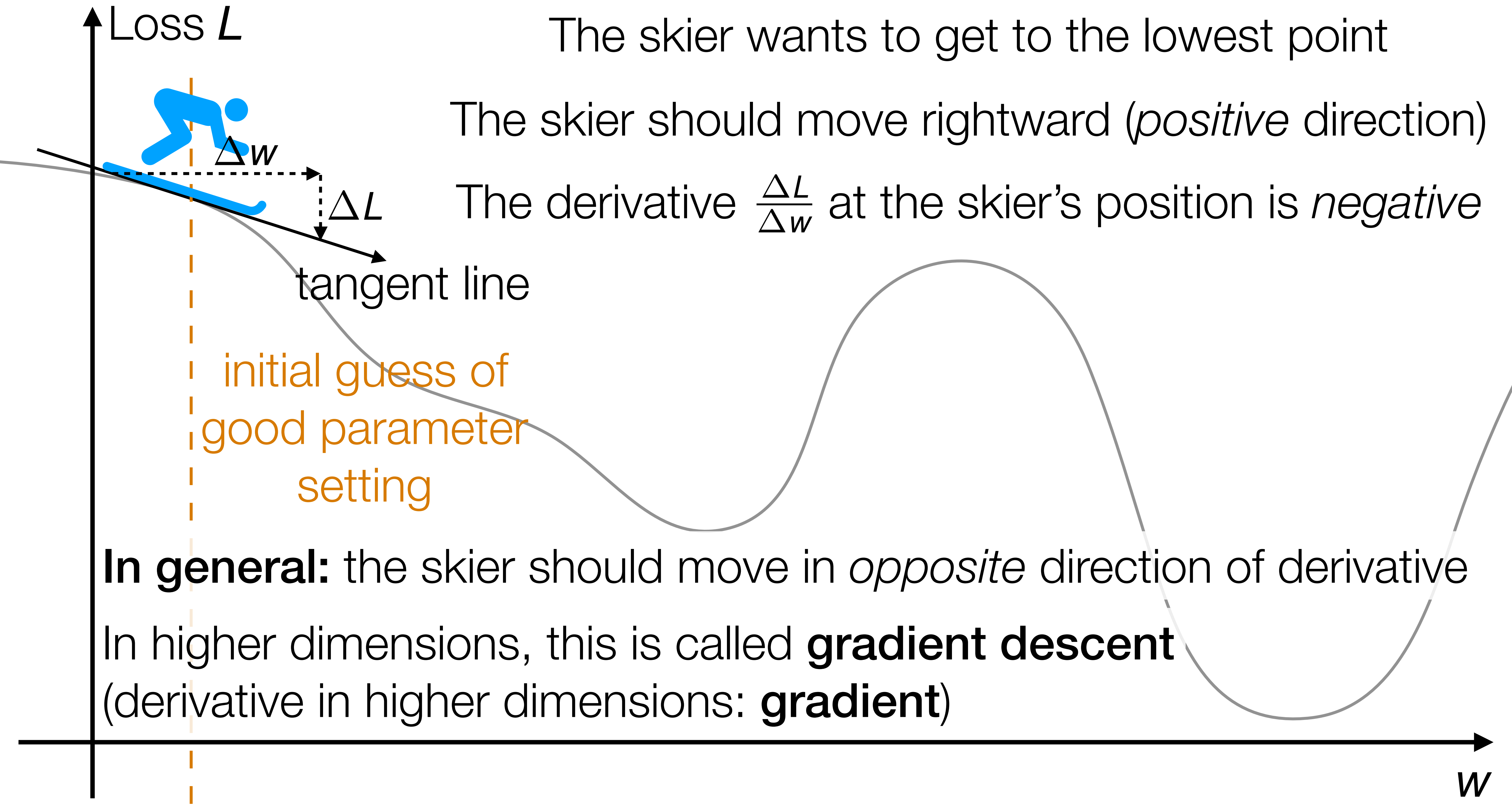
Gradient Descent

Suppose the neural network has a single real number parameter ***w***

The skier wants to get to the lowest point

The skier should move rightward (*positive* direction)

The derivative $\frac{\Delta L}{\Delta w}$ at the skier's position is *negative*

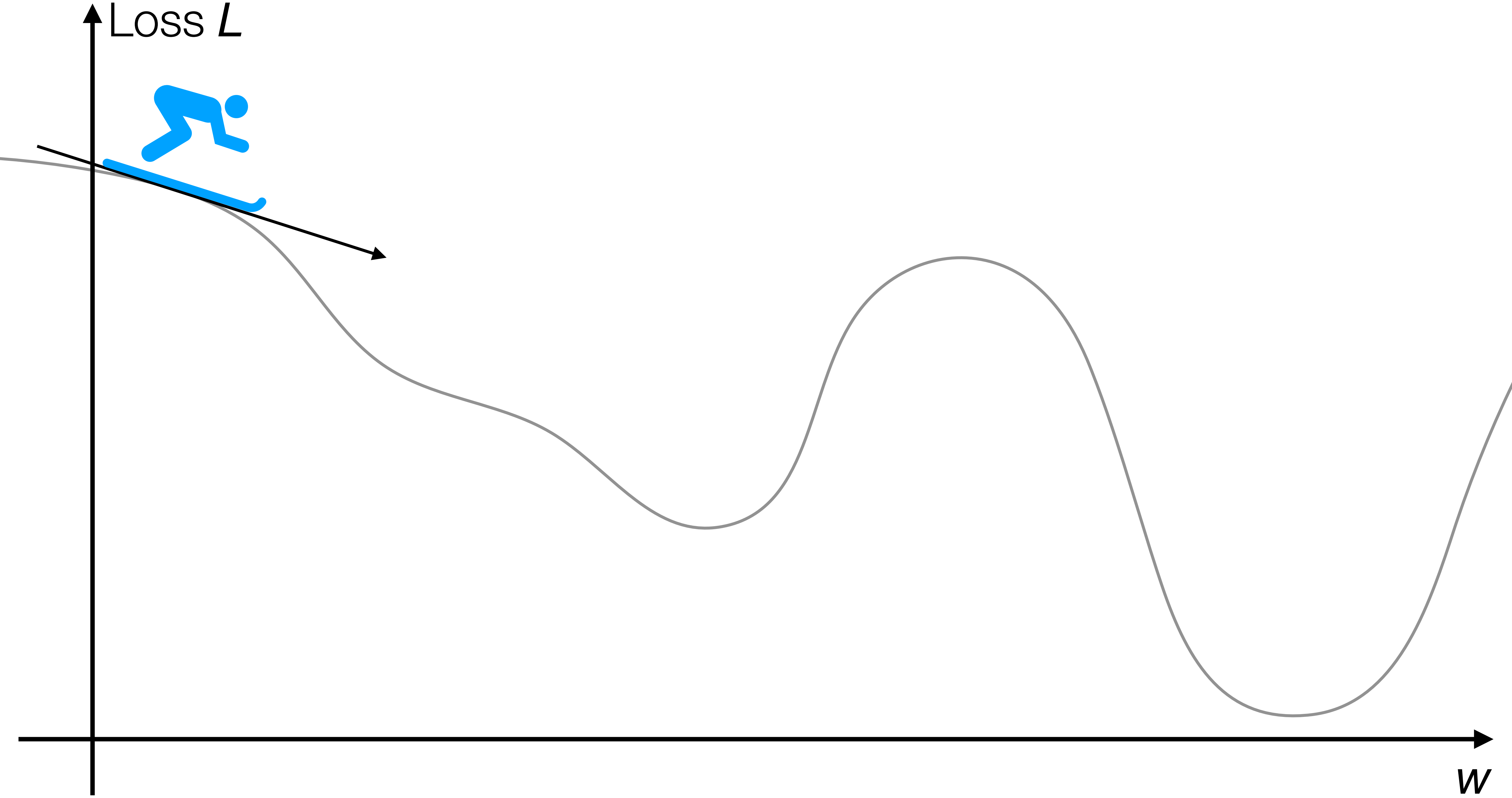


In general: the skier should move in *opposite* direction of derivative

In higher dimensions, this is called **gradient descent**
(derivative in higher dimensions: **gradient**)

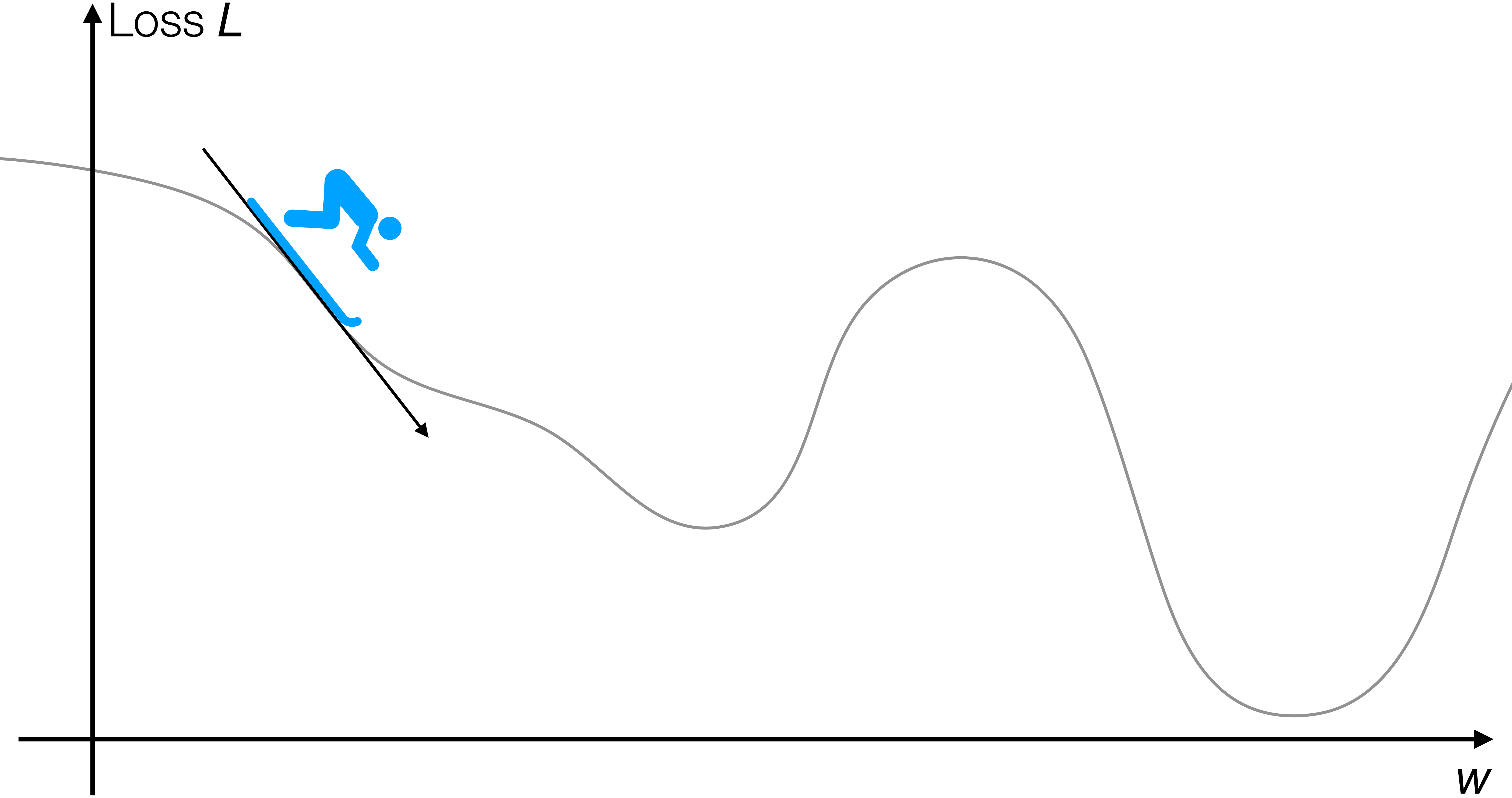
Gradient Descent

Suppose the neural network has a single real number parameter *w*



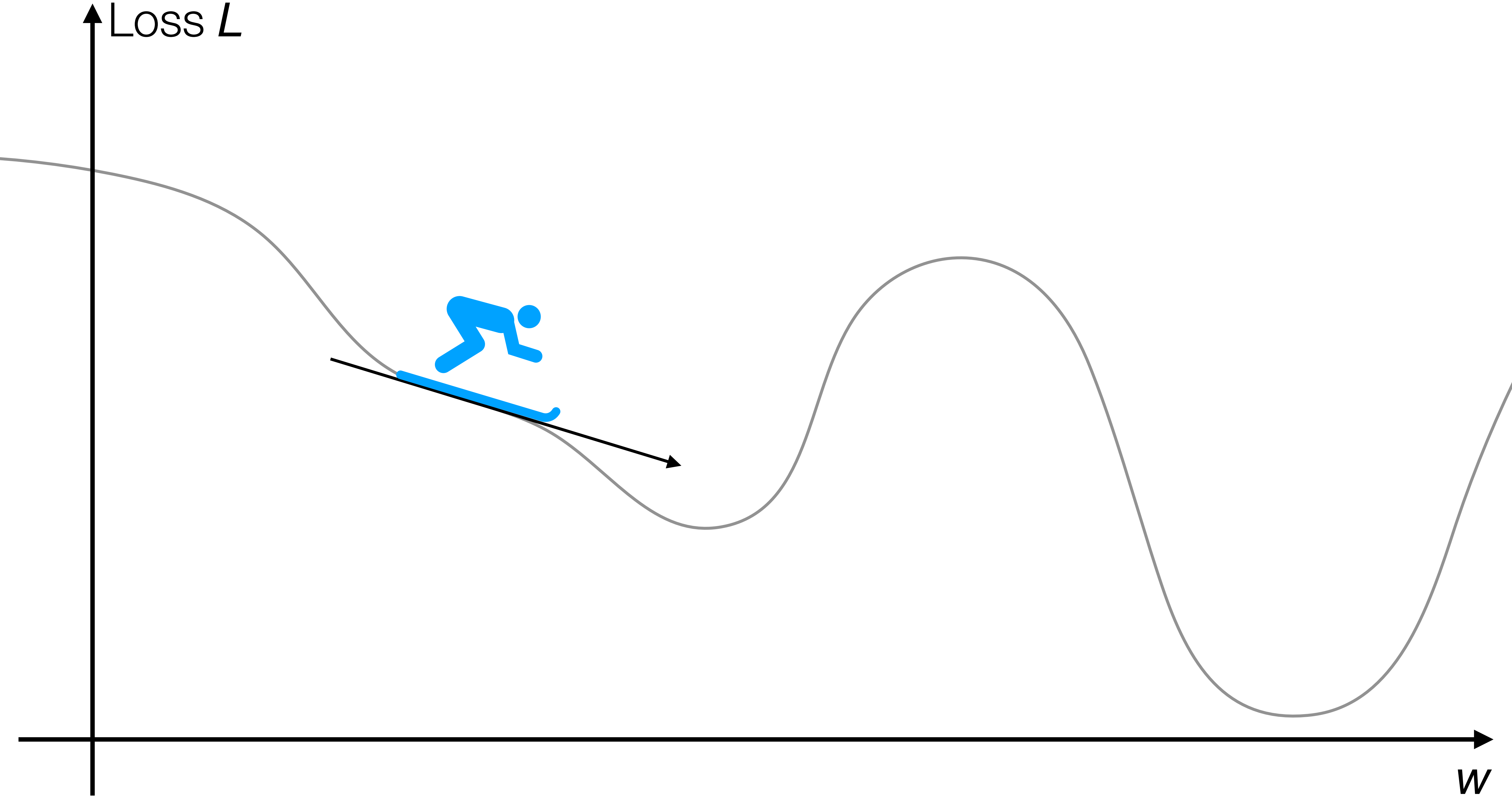
Gradient Descent

Suppose the neural network has a single real number parameter *w*



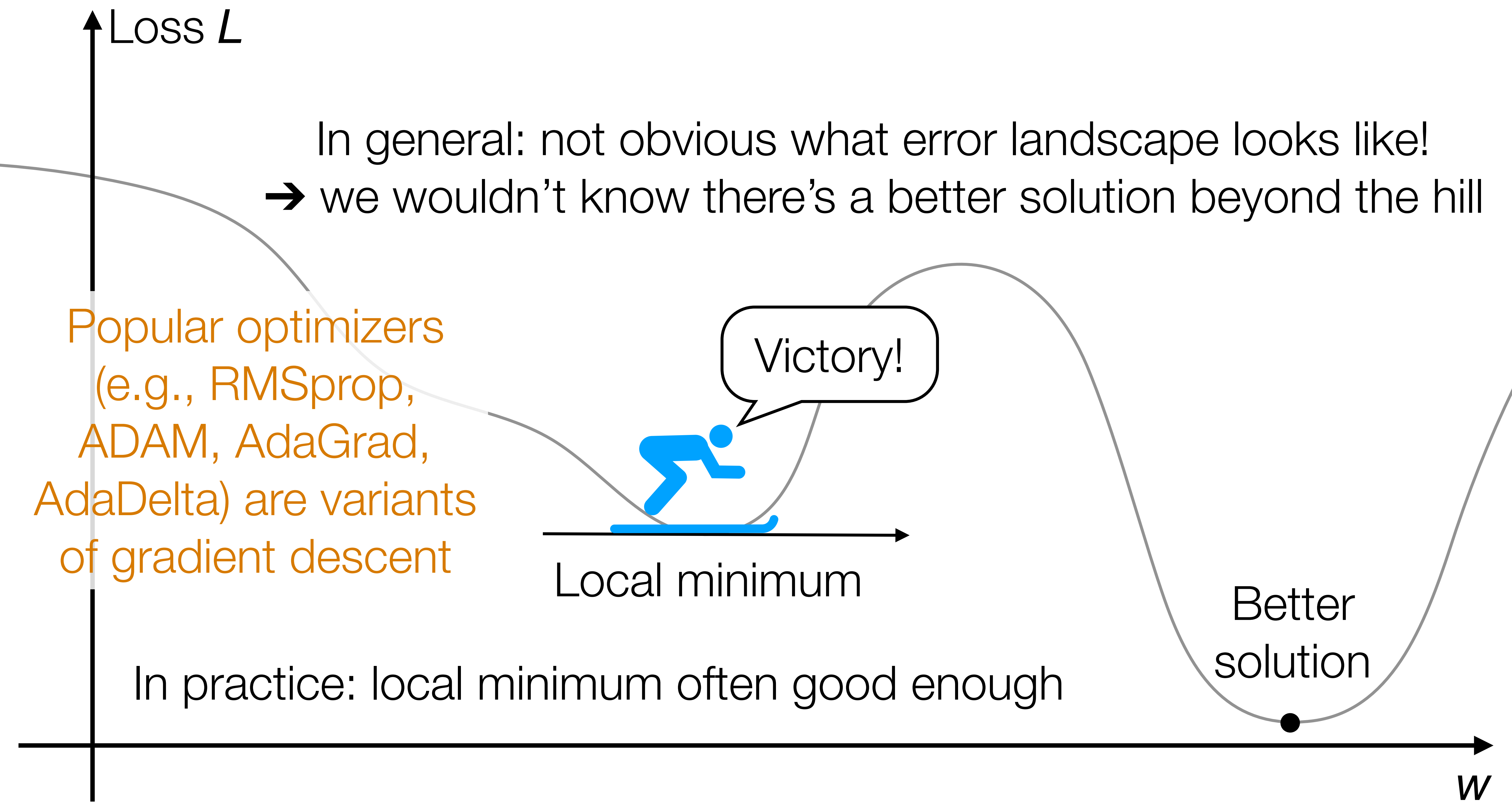
Gradient Descent

Suppose the neural network has a single real number parameter *w*



Gradient Descent

Suppose the neural network has a single real number parameter *w*



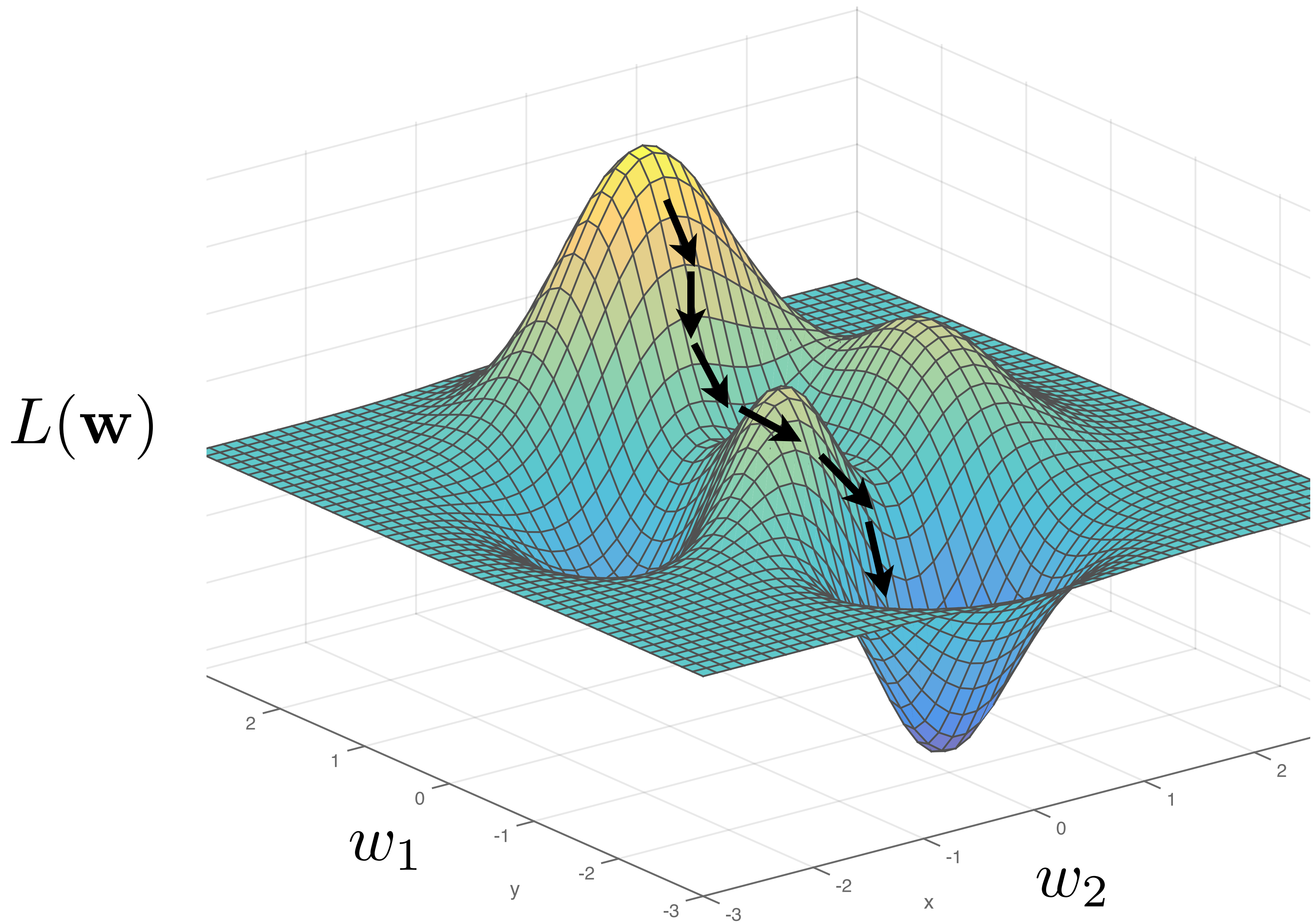
In general: not obvious what error landscape looks like!
➔ we wouldn't know there's a better solution beyond the hill

Popular optimizers (e.g., RMSprop, ADAM, AdaGrad, AdaDelta) are variants of gradient descent

In practice: local minimum often good enough

Gradient Descent

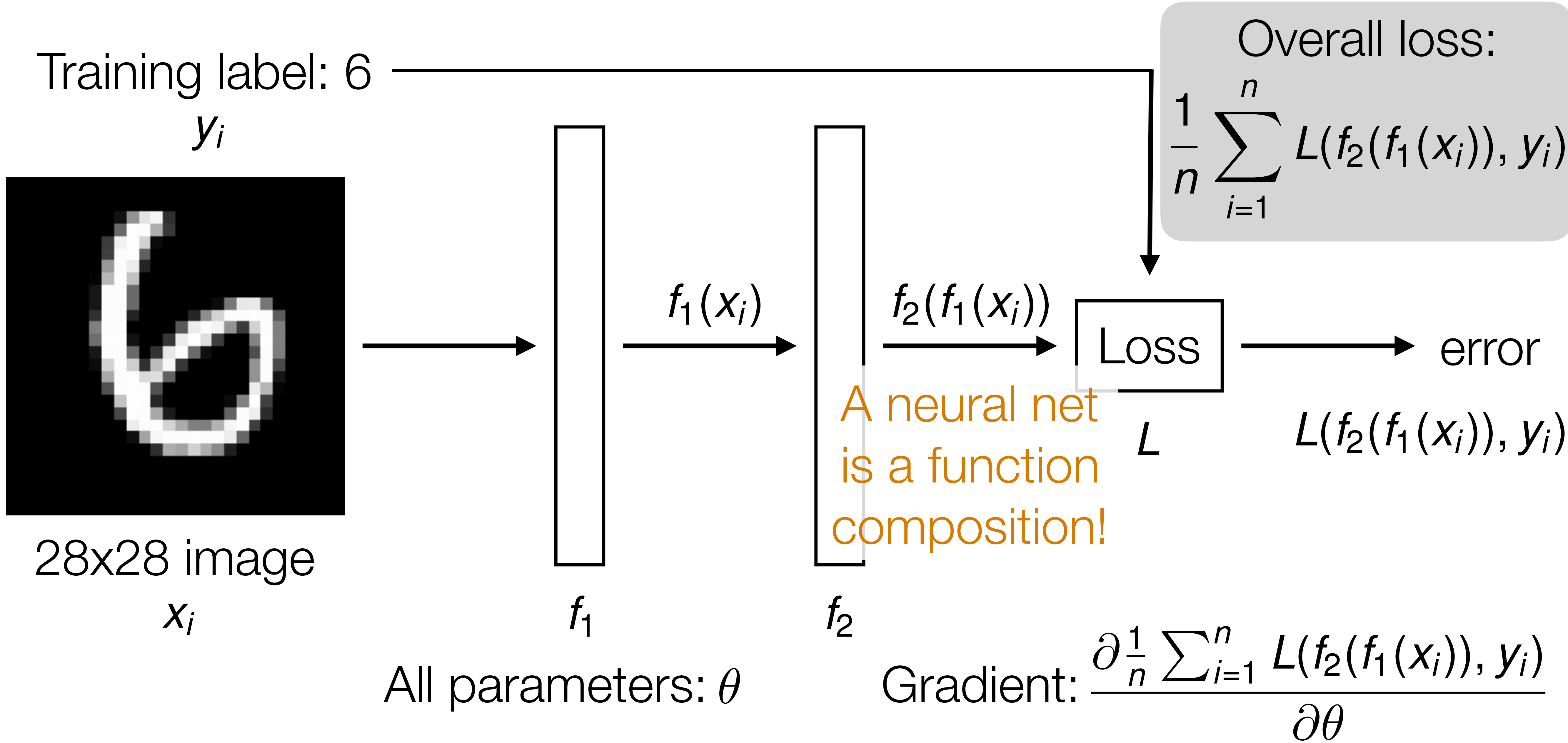
2D example



Slide by Phillip Isola

Remark: In practice, deep nets often have > *millions* of parameters, so *very* high-dimensional gradient descent

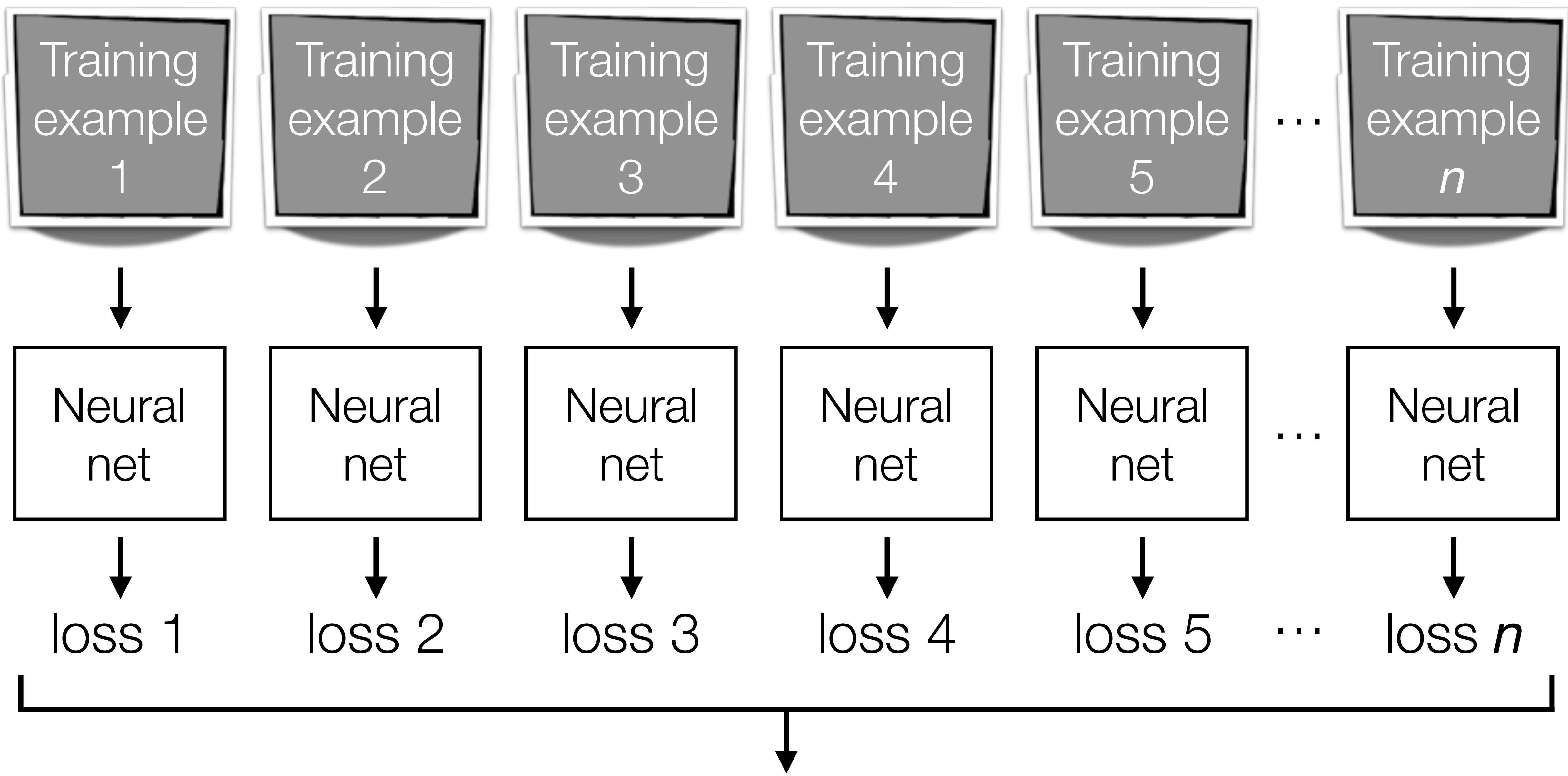
Handwritten Digit Recognition



Automatic differentiation is crucial in learning deep nets!

Careful derivative chain rule calculation: **back-propagation**

Gradient Descent



We have to compute lots of gradients to help the skier know where to go!

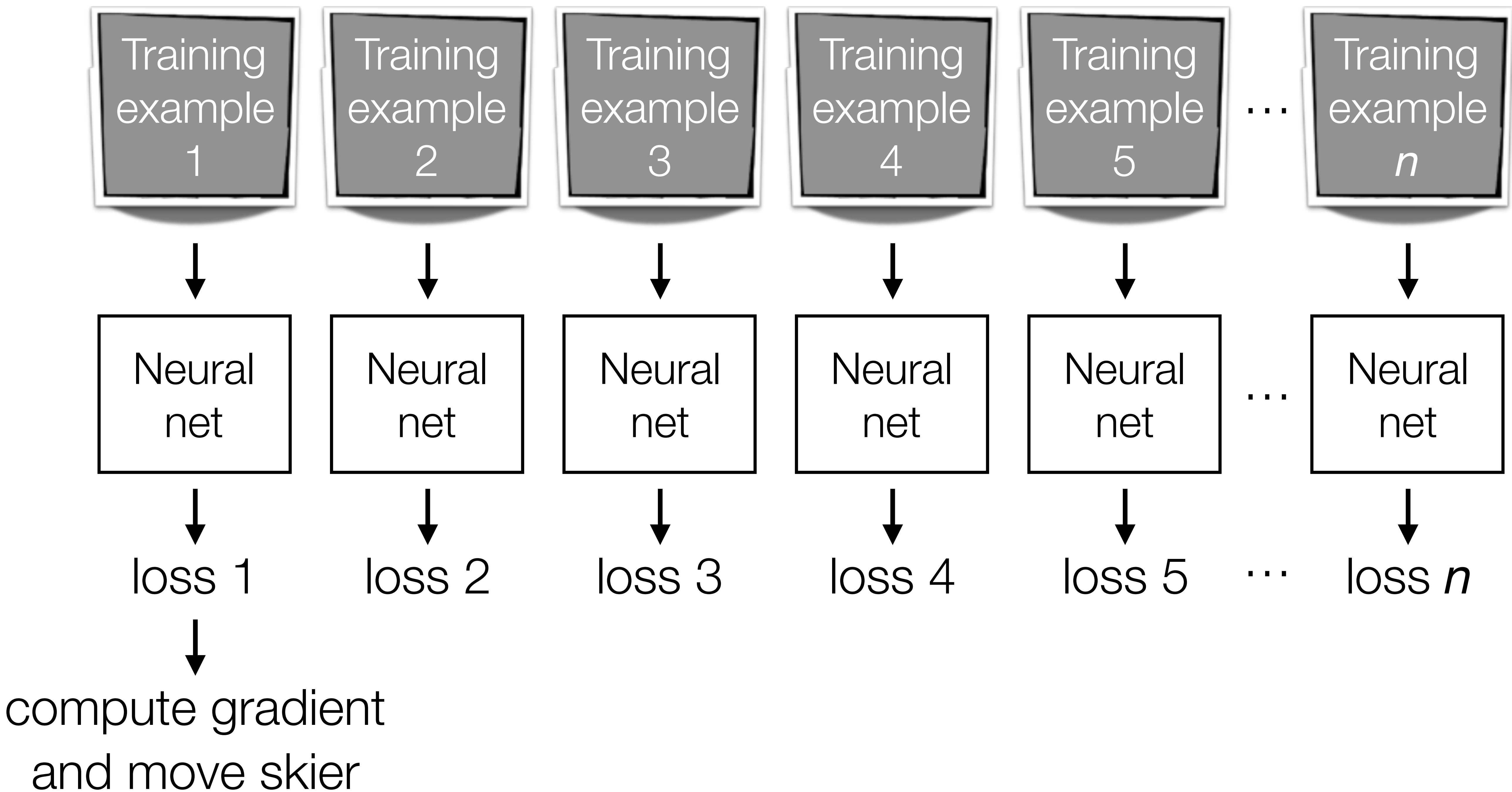
average loss

↓

compute gradient and move skier

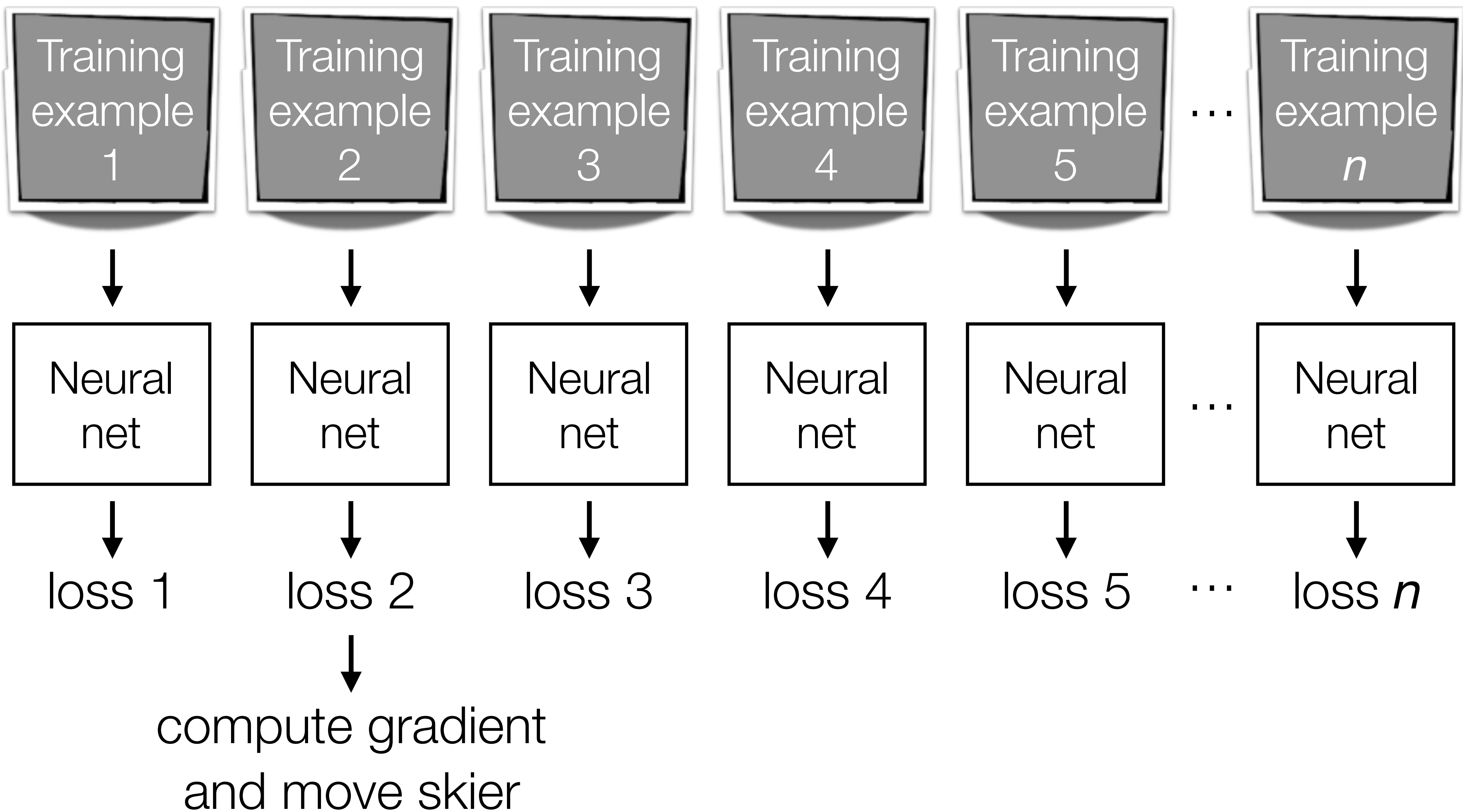
Computing gradients using all the training data seems really expensive!

Stochastic Gradient Descent (SGD)



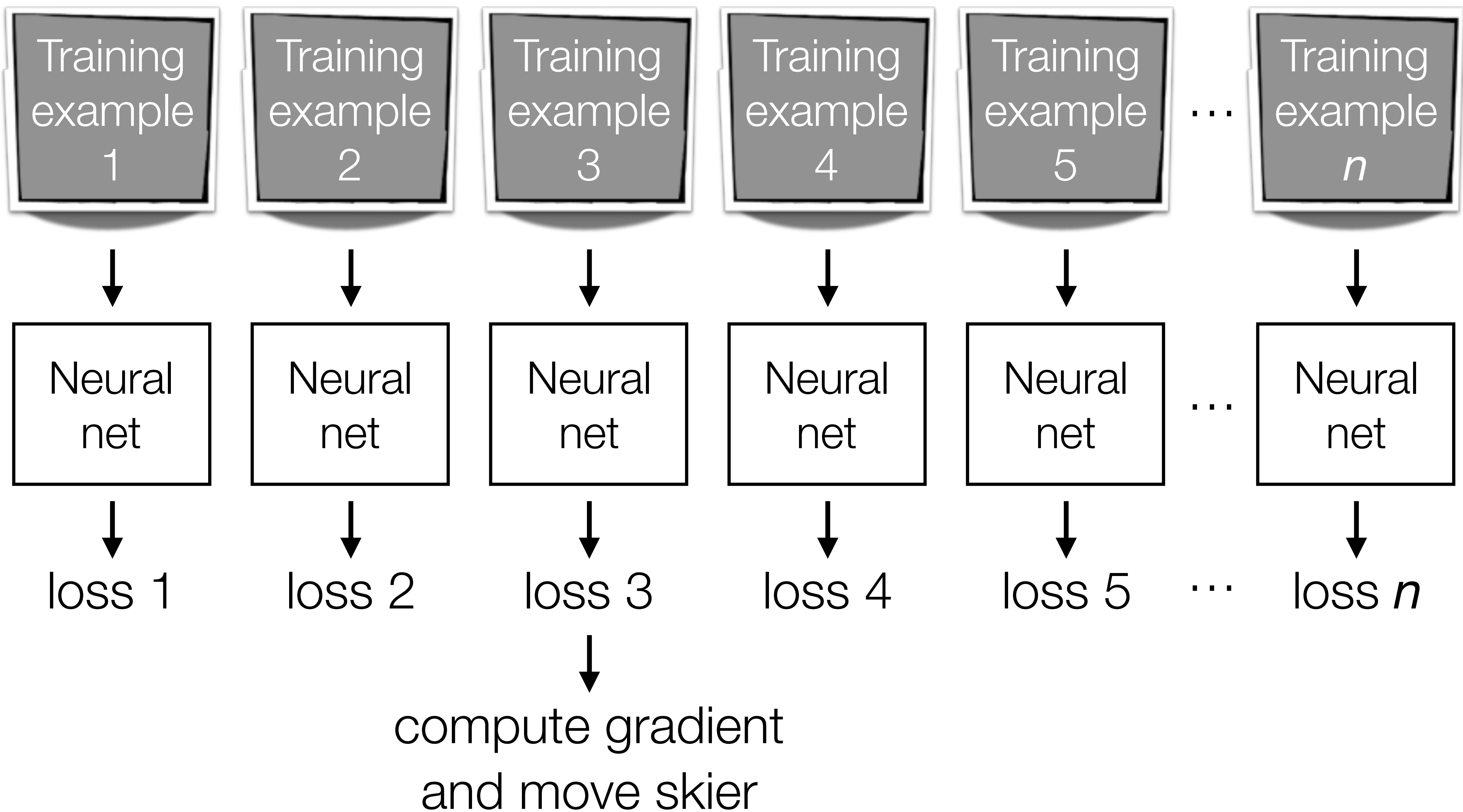
SGD: compute gradient using only 1 training example at a time
(can think of this gradient as a noisy approximation of the "full" gradient)

Stochastic Gradient Descent (SGD)



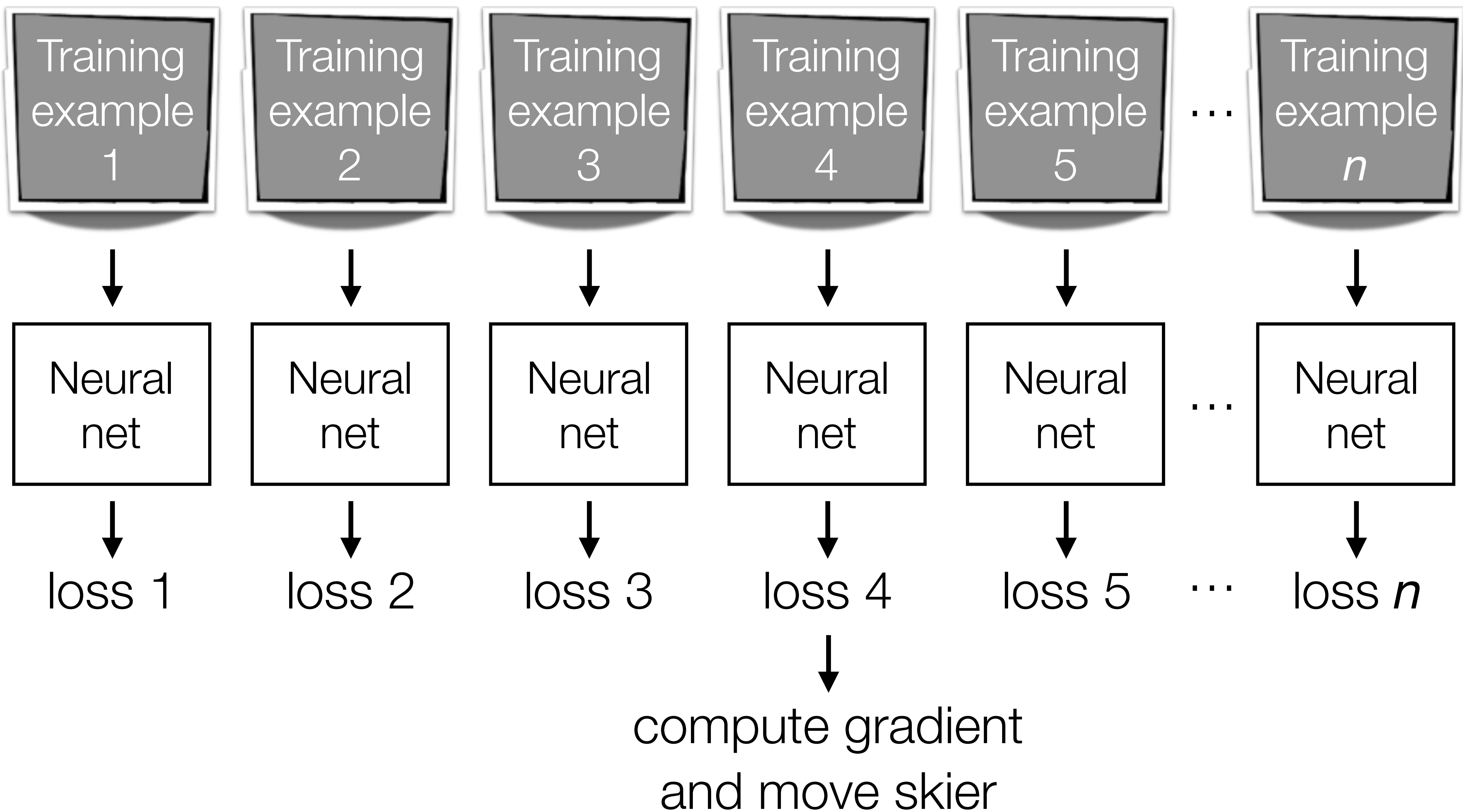
SGD: compute gradient using only 1 training example at a time
(can think of this gradient as a noisy approximation of the "full" gradient)

Stochastic Gradient Descent (SGD)



SGD: compute gradient using only 1 training example at a time
(can think of this gradient as a noisy approximation of the "full" gradient)

Stochastic Gradient Descent (SGD)



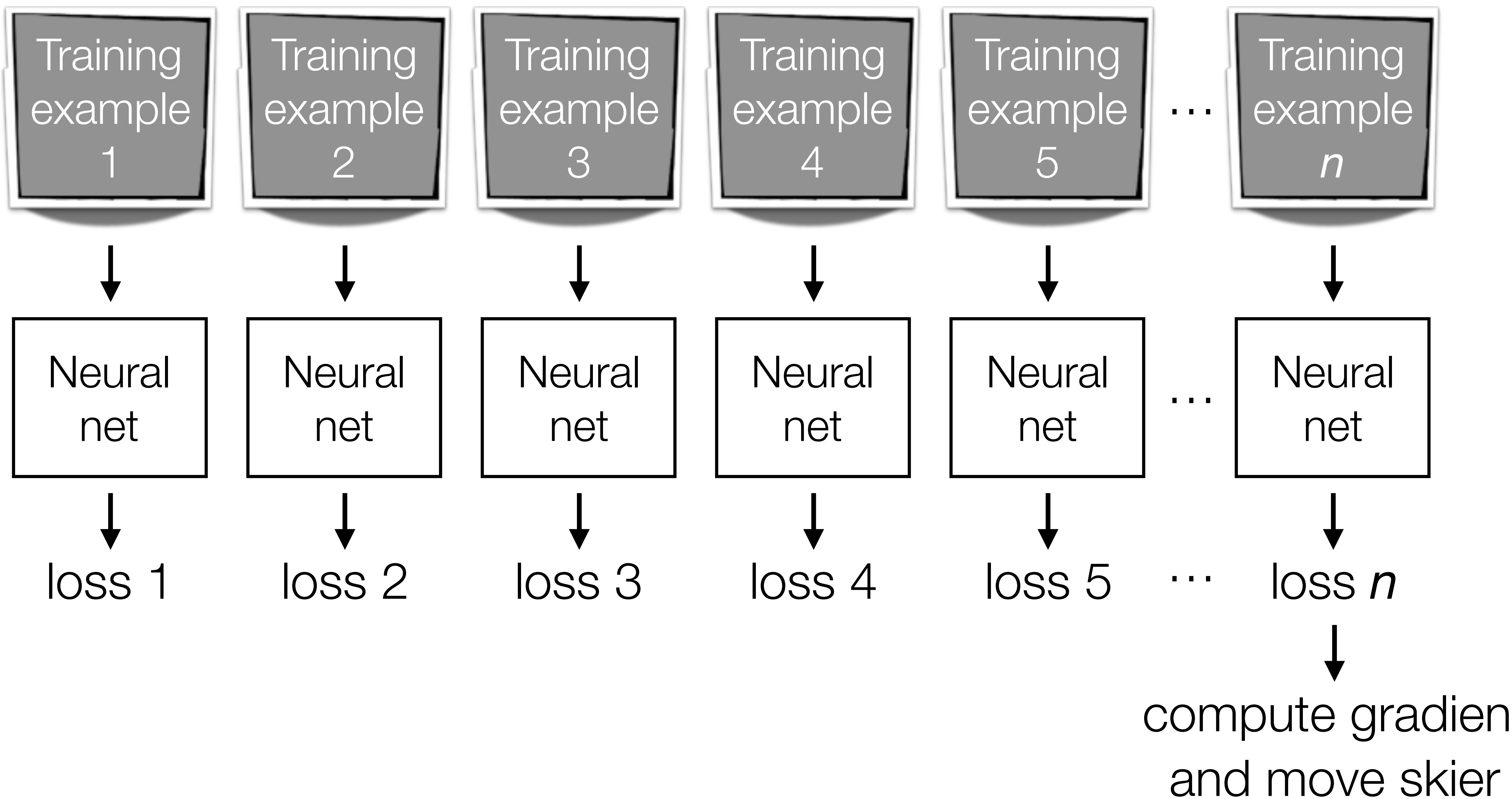
SGD: compute gradient using only 1 training example at a time
(can think of this gradient as a noisy approximation of the “full” gradient)

Stochastic Gradient Descent (SGD)

Training example 1	Training example 2	Training example 3	Training example 4	Training example 5	…	Training example *n*
↓	↓	↓	↓	↓		↓
Neural net	Neural net	Neural net	Neural net	Neural net	…	Neural net
↓	↓	↓	↓	↓		↓
loss 1	loss 2	loss 3	loss 4	loss 5	…	loss *n*
				↓		
				compute gradient and move skier		

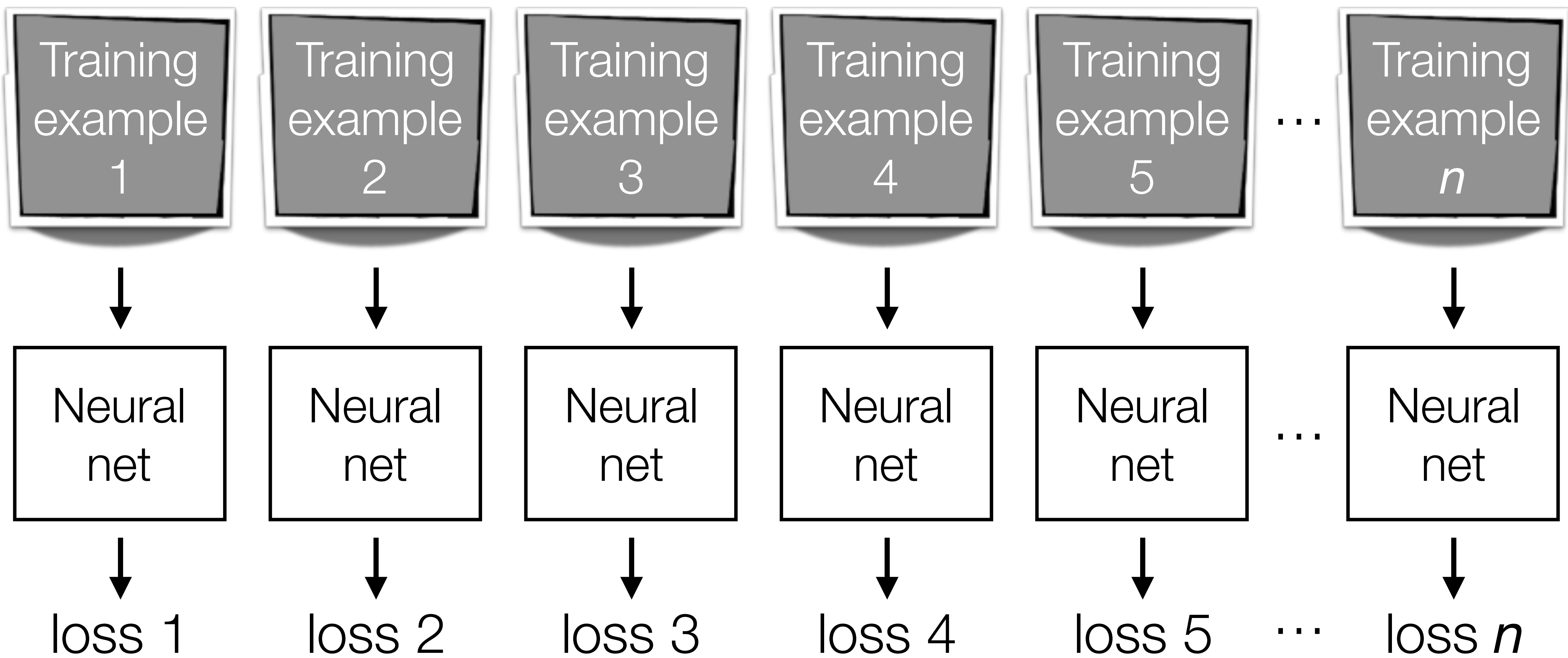
SGD: compute gradient using only 1 training example at a time
(can think of this gradient as a noisy approximation of the "full" gradient)

Stochastic Gradient Descent (SGD)



SGD: compute gradient using only 1 training example at a time
(can think of this gradient as a noisy approximation of the "full" gradient)

Stochastic Gradient Descent (SGD)

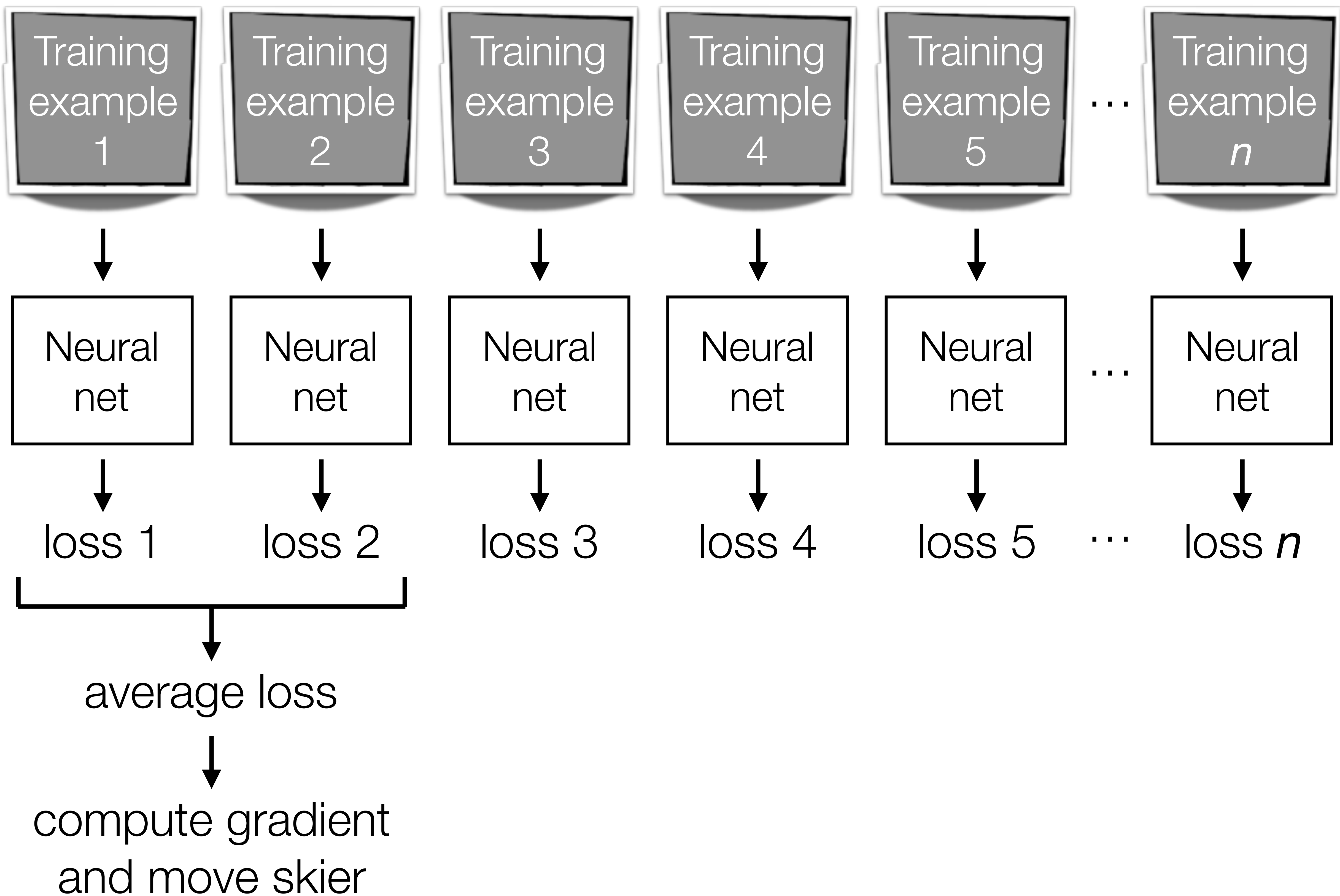


compute gradient
and move skier

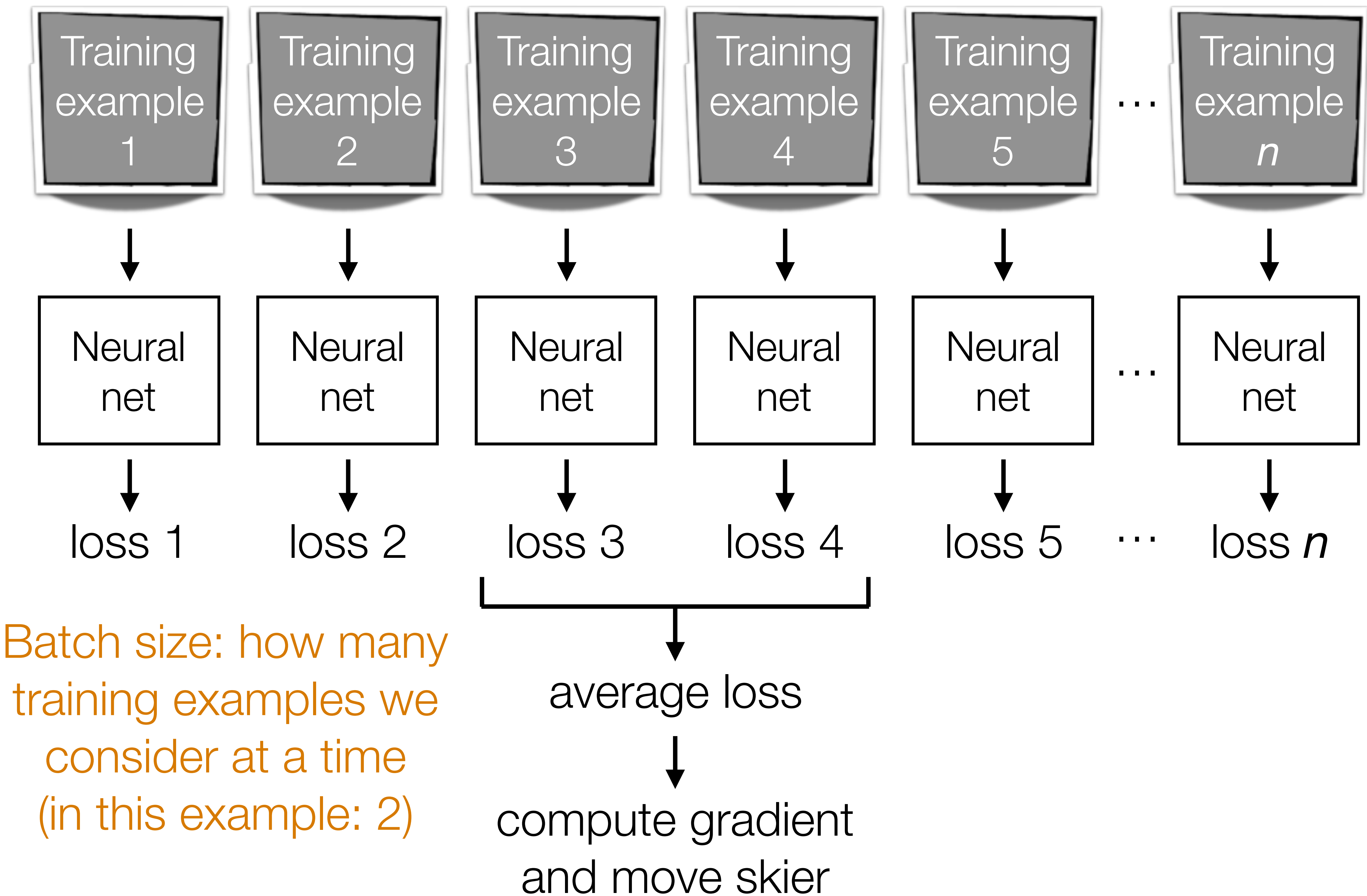
An epoch refers to 1 full pass
through all the training data

SGD: compute gradient using only 1 training example at a time
(can think of this gradient as a noisy approximation of the "full" gradient)

Mini-Batch Gradient Descent



Mini-Batch Gradient Descent



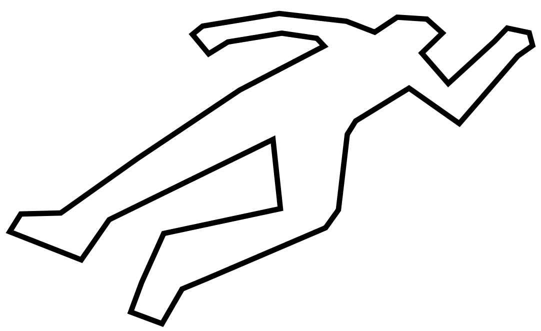
The Future of Deep Learning

- Deep learning currently is still limited in what it can do — the layers do simple operations and have to be differentiable
    - How do we make deep nets that generalize better?
- Still lots of engineering and expert knowledge used to design some of the best systems (e.g., AlphaGo)
    - How do we get away with using less expert knowledge?
- How do we do lifelong learning?

Unstructured Data Analysis

Not detailed in lecture but addressed by final project

Question



The dead body

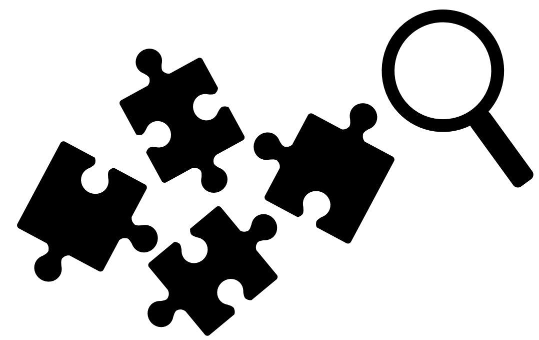
This is provided by a practitioner

Data

The evidence

Some times you have to collect more evidence!

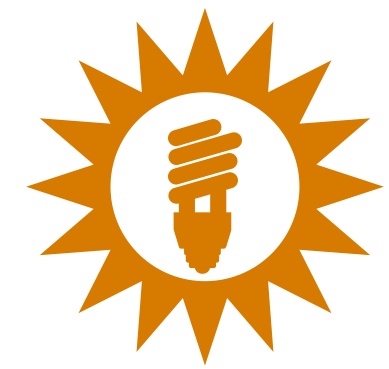
Finding Structure



Puzzle solving, careful analysis

Exploratory data analysis

Insights



When? Where? Why? How? Perpetrator catchable?

Answer original question

There isn't always a follow-up prediction problem to solve!

UDA involves *lots* of data ➔ write computer programs to assist analysis

94-775 Some Parting Thoughts

- Remember to **visualize different steps of your data analysis pipeline**
  - Helpful for both debugging and interpreting final output!
- Very often there are *tons* of models/design choices to try
  - Come up with **quantitative metrics** that make sense for your problem, and use these metrics to **evaluate models with a prediction task on held-out data**
- Often times you won't have labels!
  - Manually obtain labels (either you do it or crowdsource)
  - Set up self-supervised learning task