Carnegie Mellon University
Heinz College

Deep Learning and 95-865 Wrap-Up

nearly all slides by George Chen (CMU)

1 slide by Phillip Isola (OpenAI, UC Berkeley)

CMU 95-865 Fall 2017

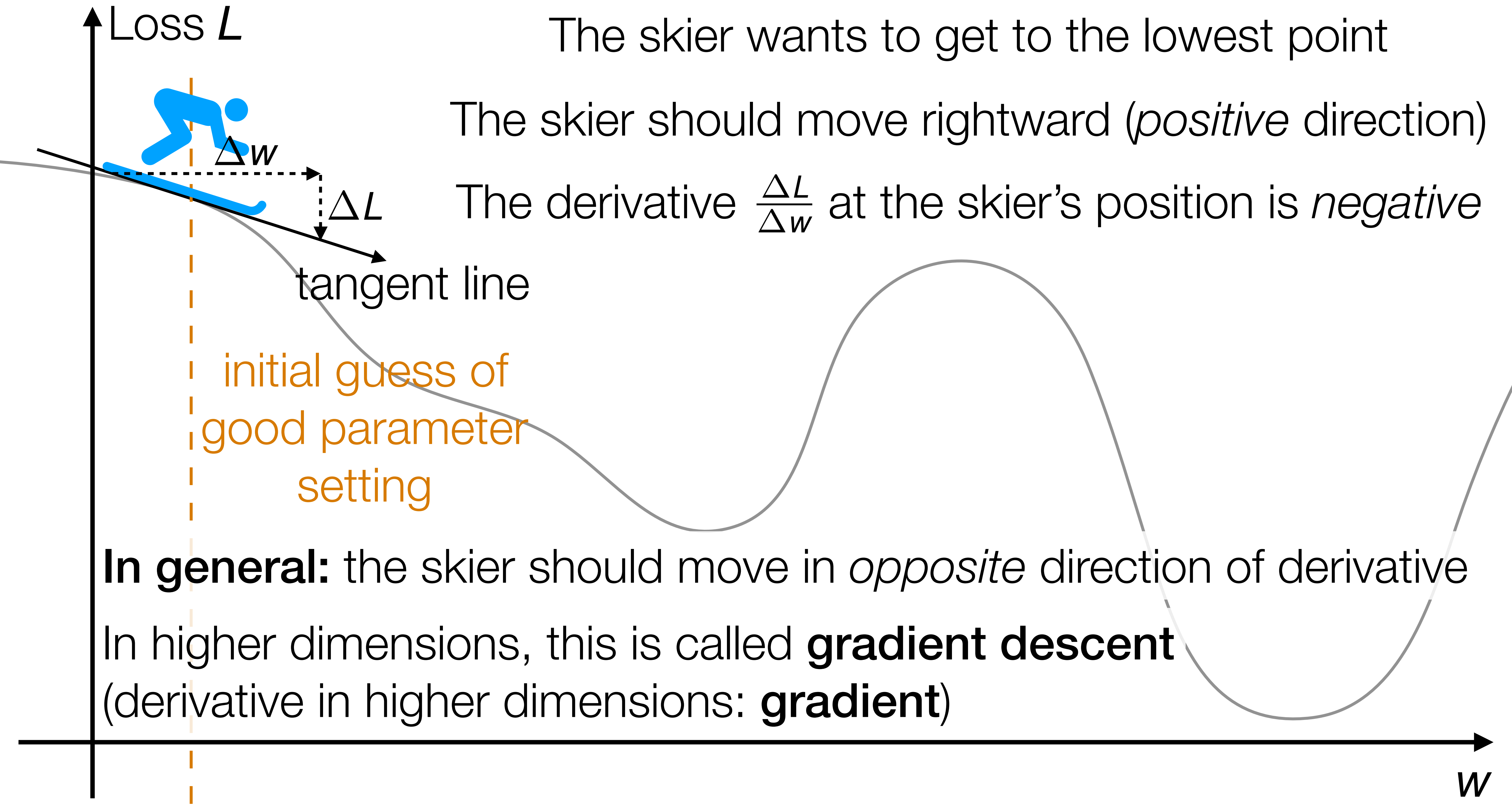
Today

- How learning a deep net works
- A bunch of deep learning topics we didn’t cover
- Course wrap-up

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)

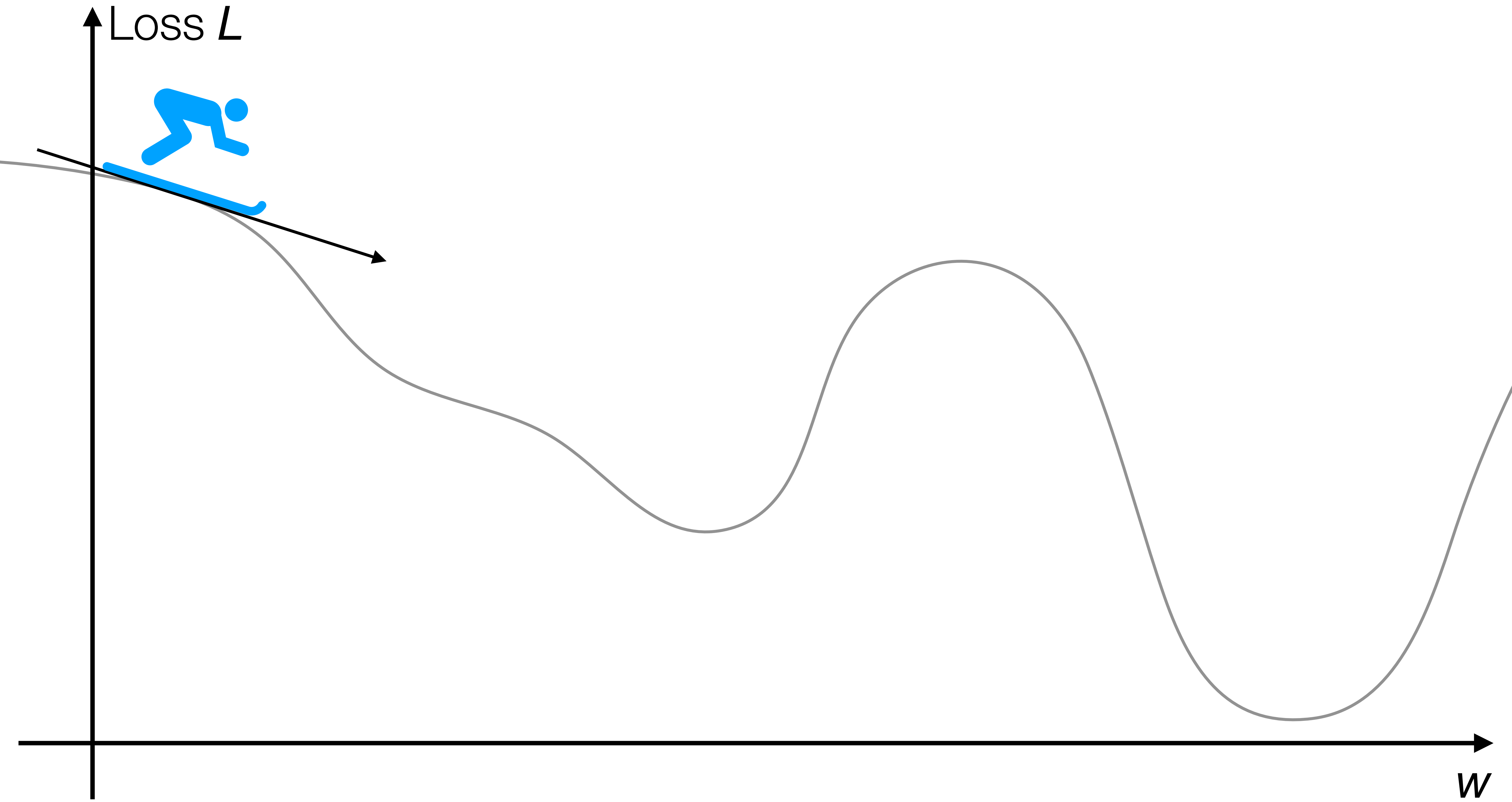
The derivative $\frac{\Delta L}{\Delta w}$ at the skier's position is *negative*

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

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

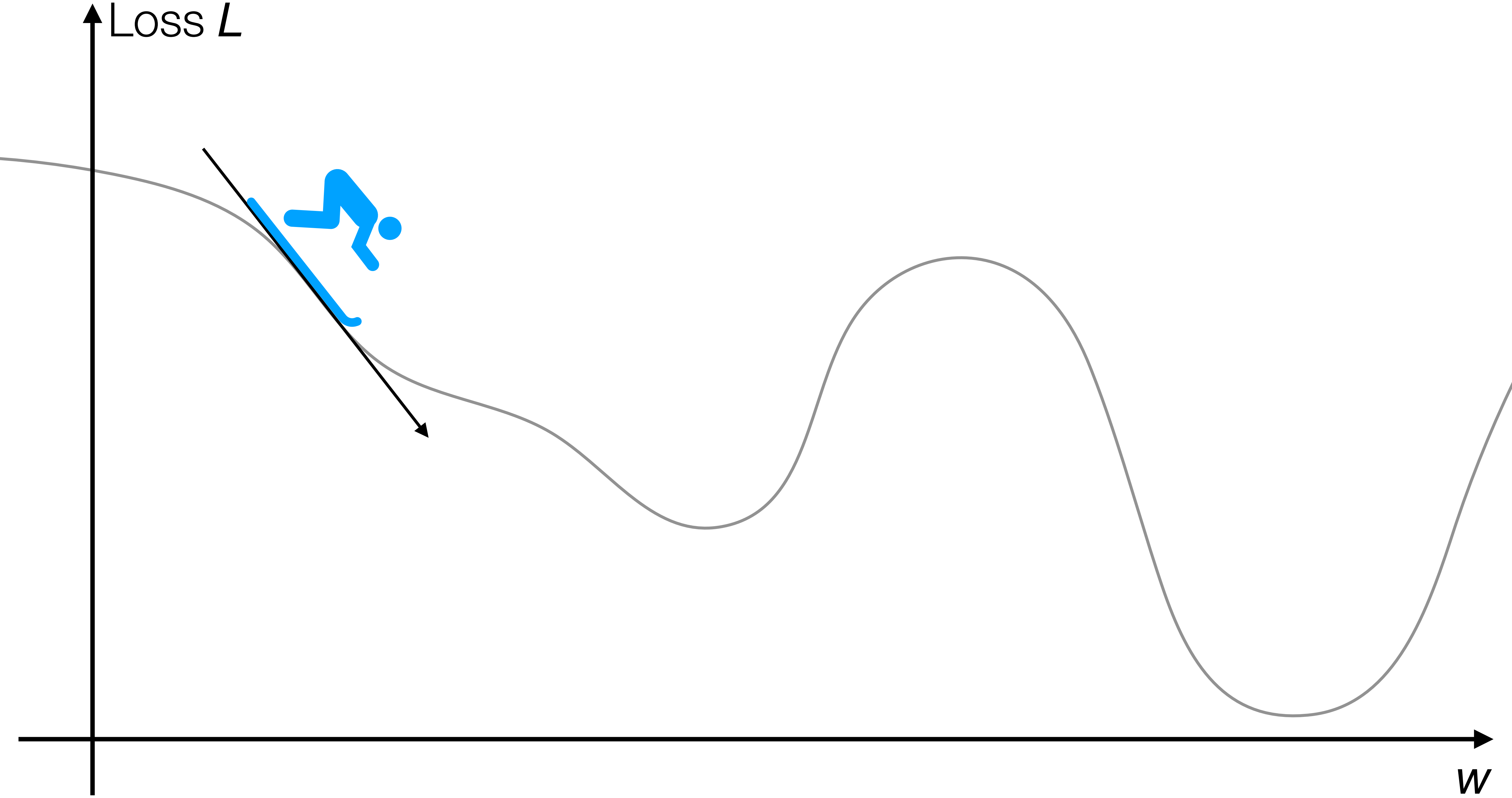
Gradient Descent

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



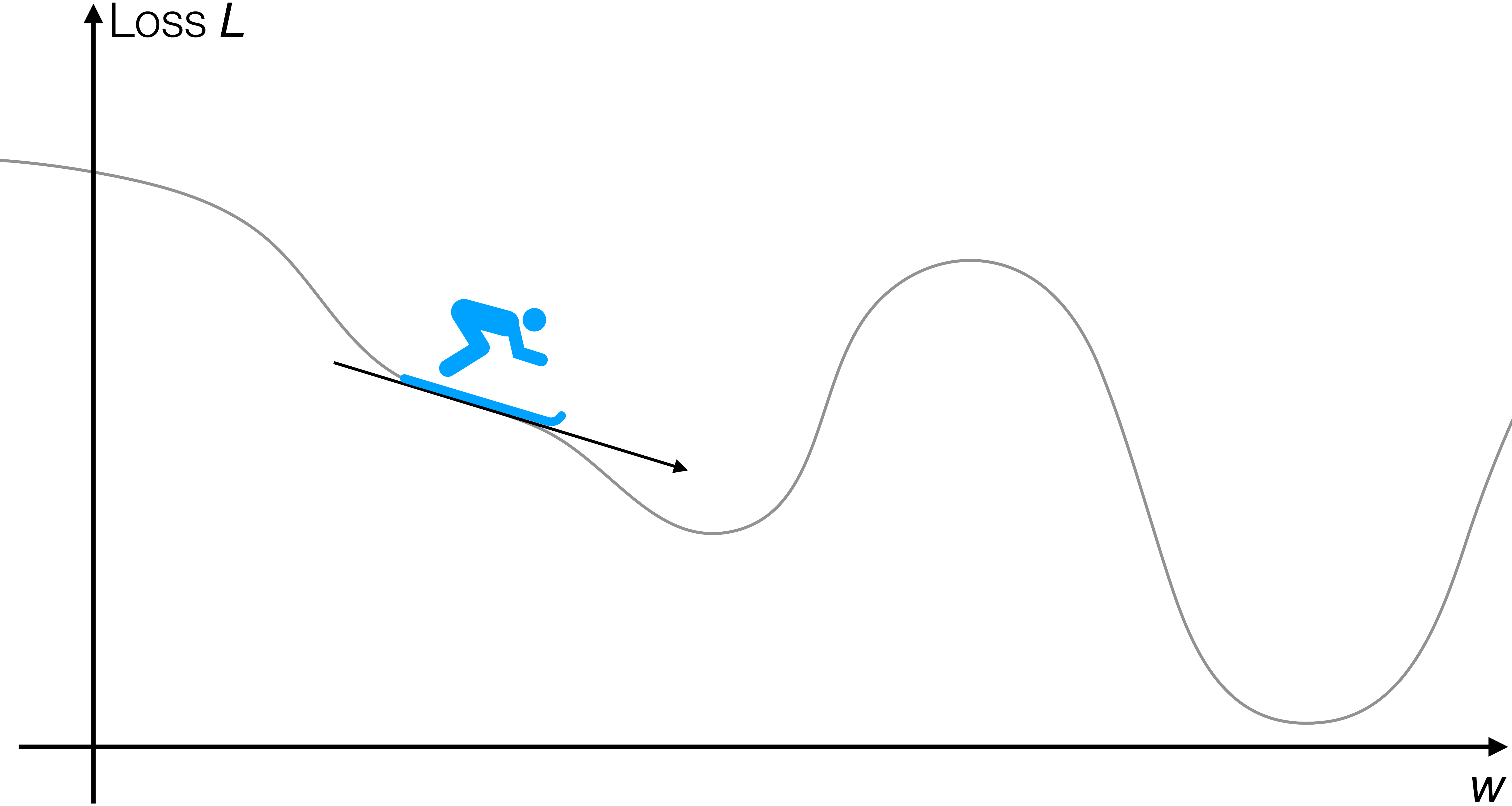
Gradient Descent

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



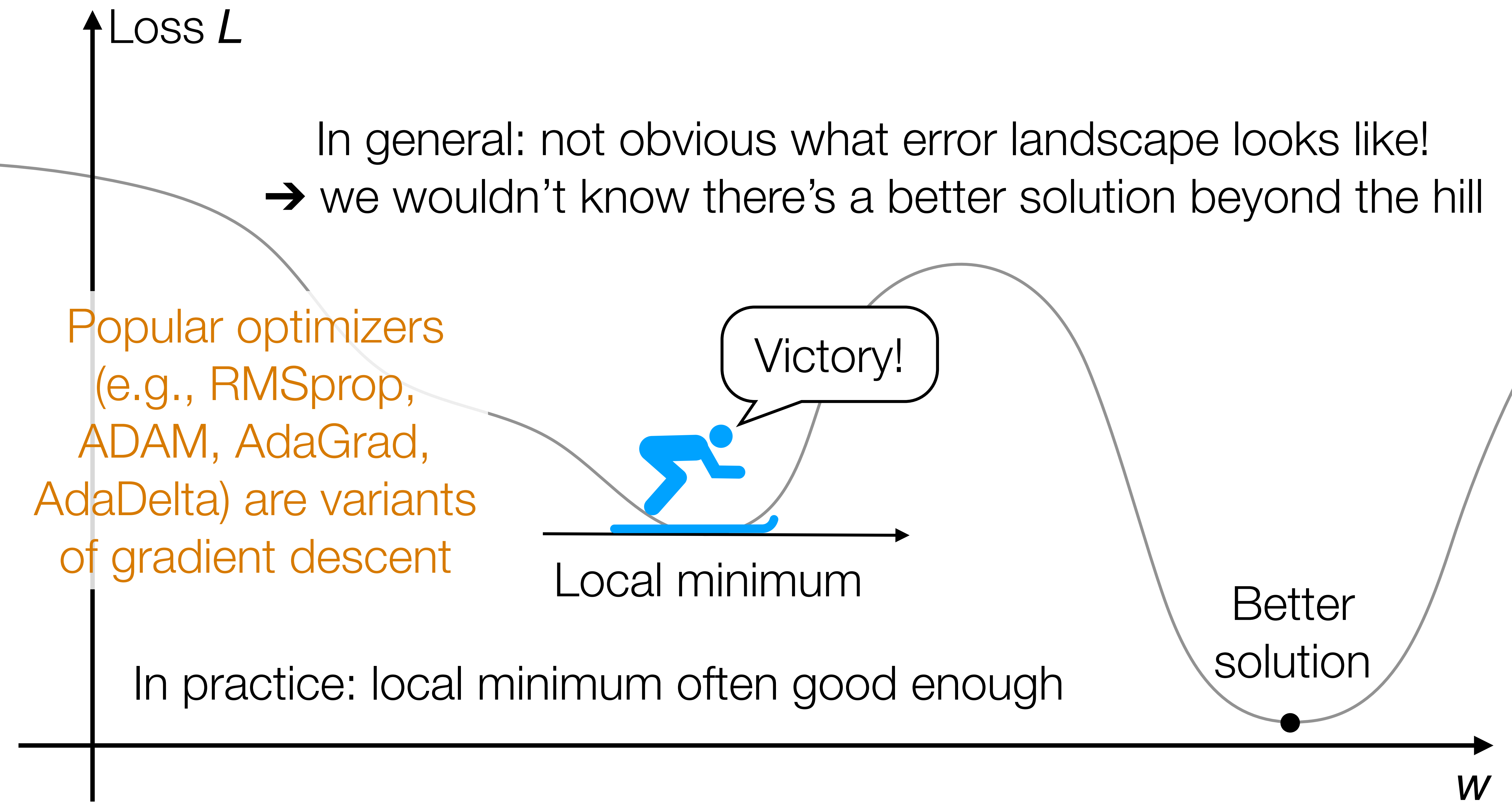
Gradient Descent

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



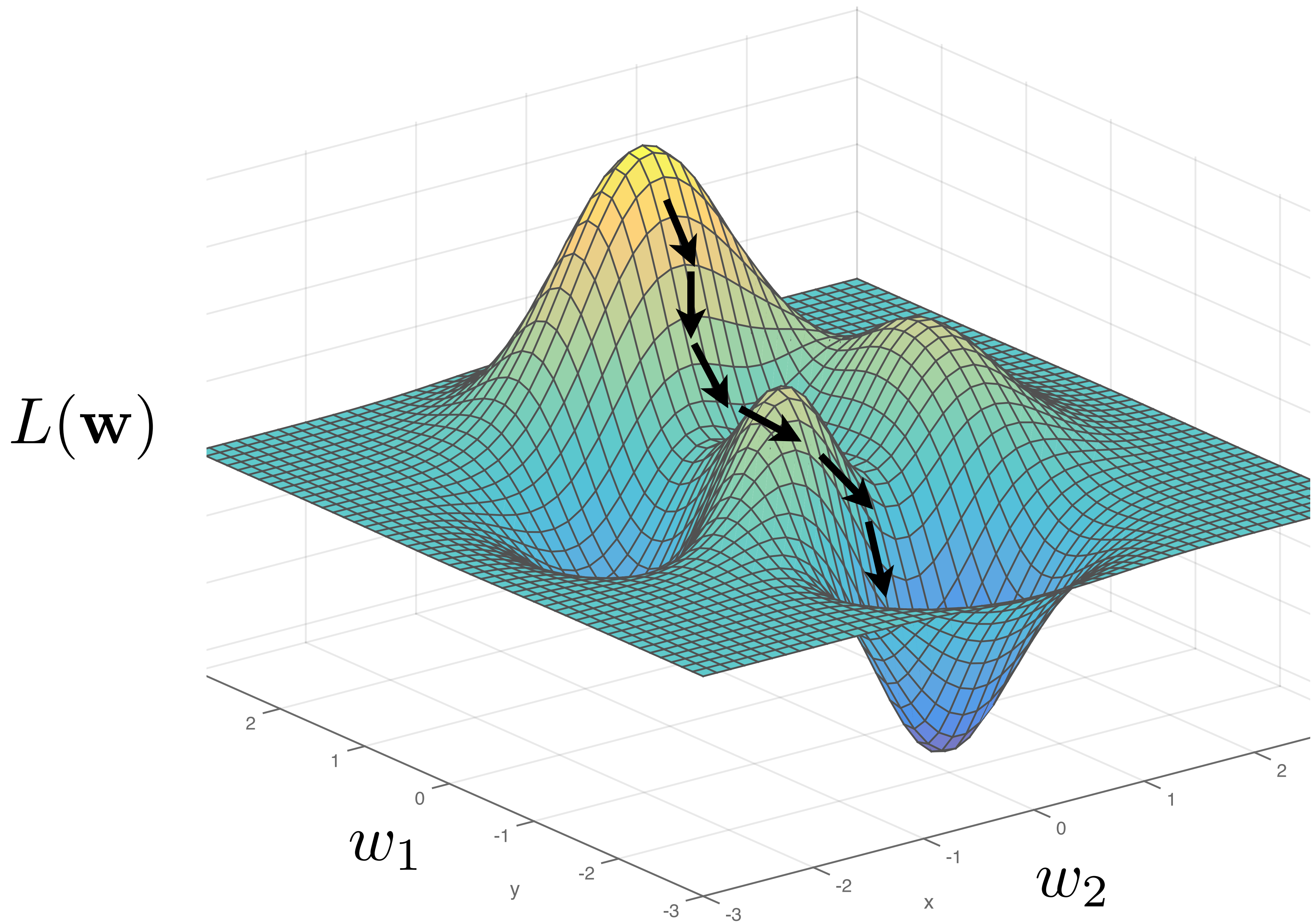
Gradient Descent

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



Gradient Descent

2D example



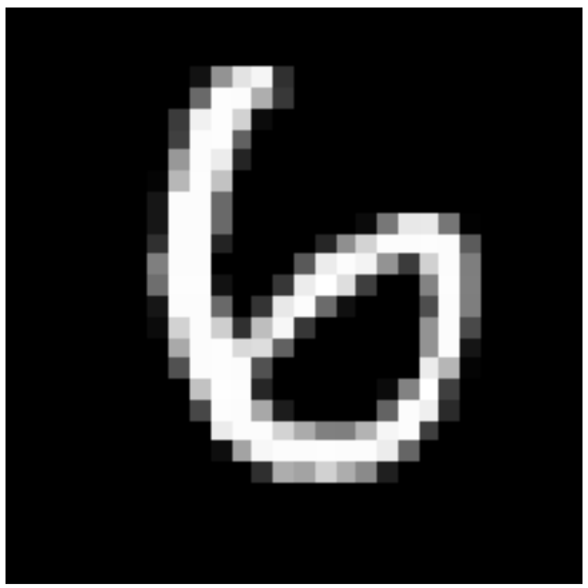
Slide by Phillip Isola

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

Handwritten Digit Recognition

Training label: 6

y_i



28x28 image

x_i

f_1 $\quad f_1(x_i)$ $\quad f_2$ $\quad f_2(f_1(x_i))$ $\quad$ Loss $\quad L$ $\quad$ error

$L(f_2(f_1(x_i)), y_i)$

Overall loss:

$$\frac{1}{n}\sum_{i=1}^{n} L(f_2(f_1(x_i)), y_i)$$

A neural net is a function composition!

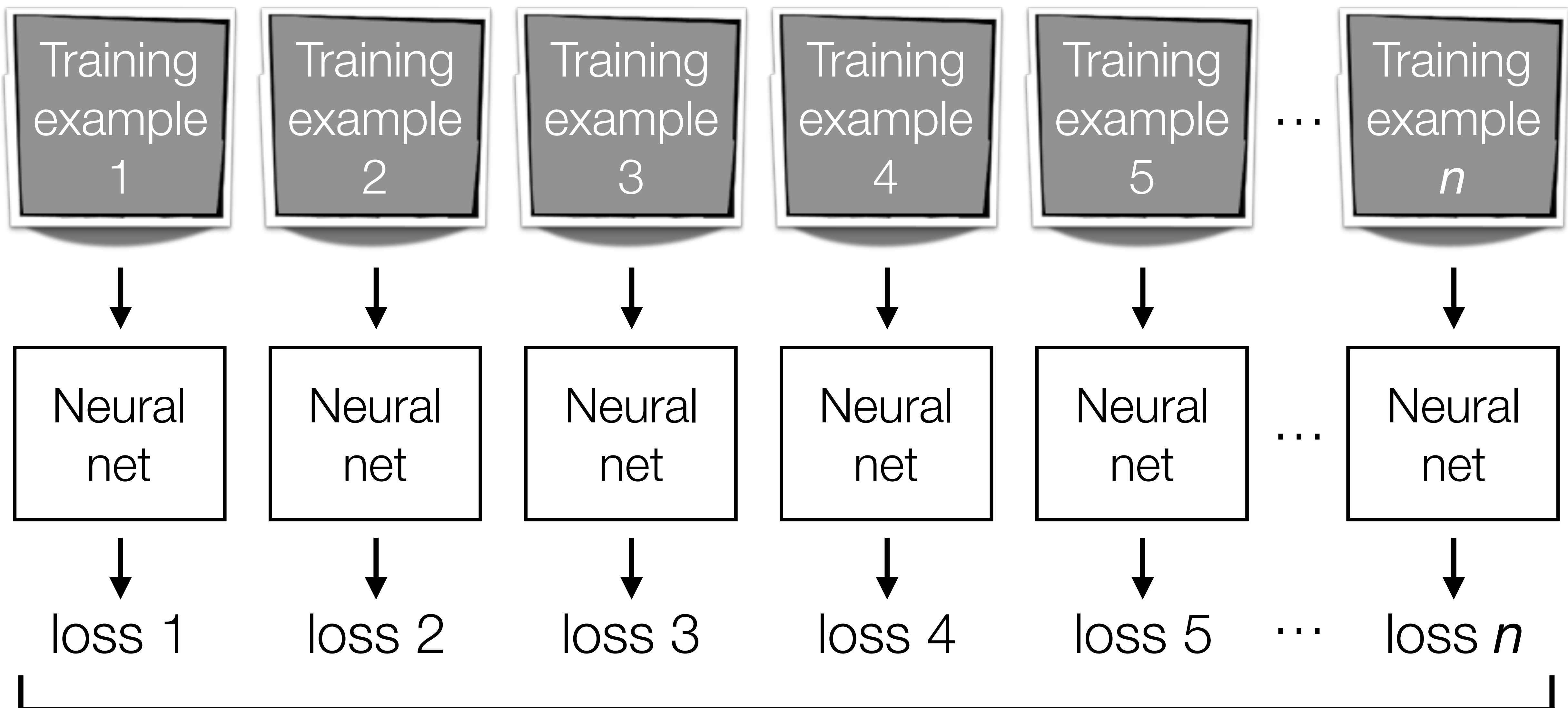
All parameters: θ

Gradient: $\dfrac{\partial \frac{1}{n}\sum_{i=1}^{n} L(f_2(f_1(x_i)), y_i)}{\partial \theta}$

Automatic differentiation is crucial in learning deep nets!

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

Gradient Descent



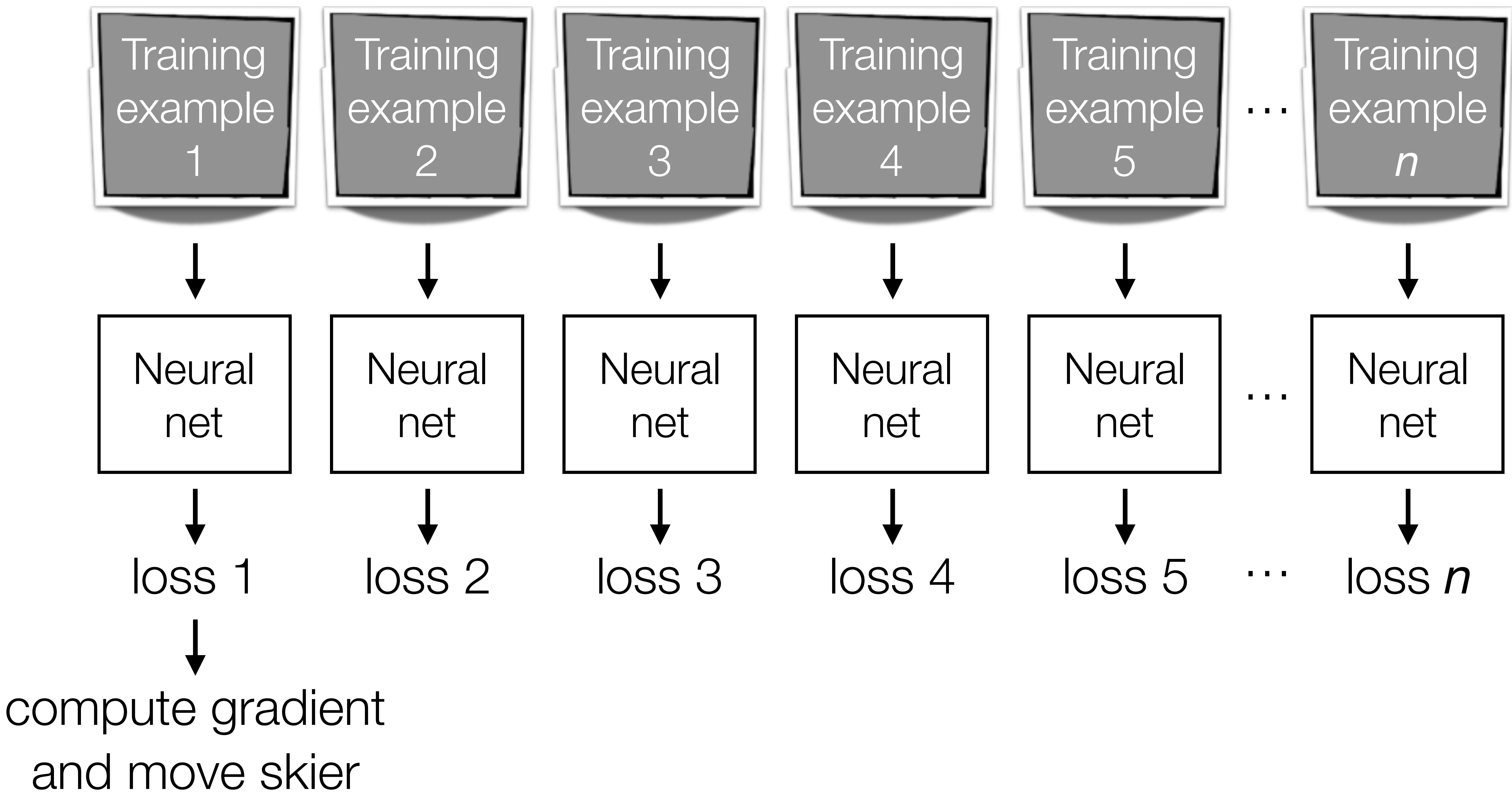
We have to compute lots of gradients to help the skier know where to go!

average loss

compute gradient and move skier

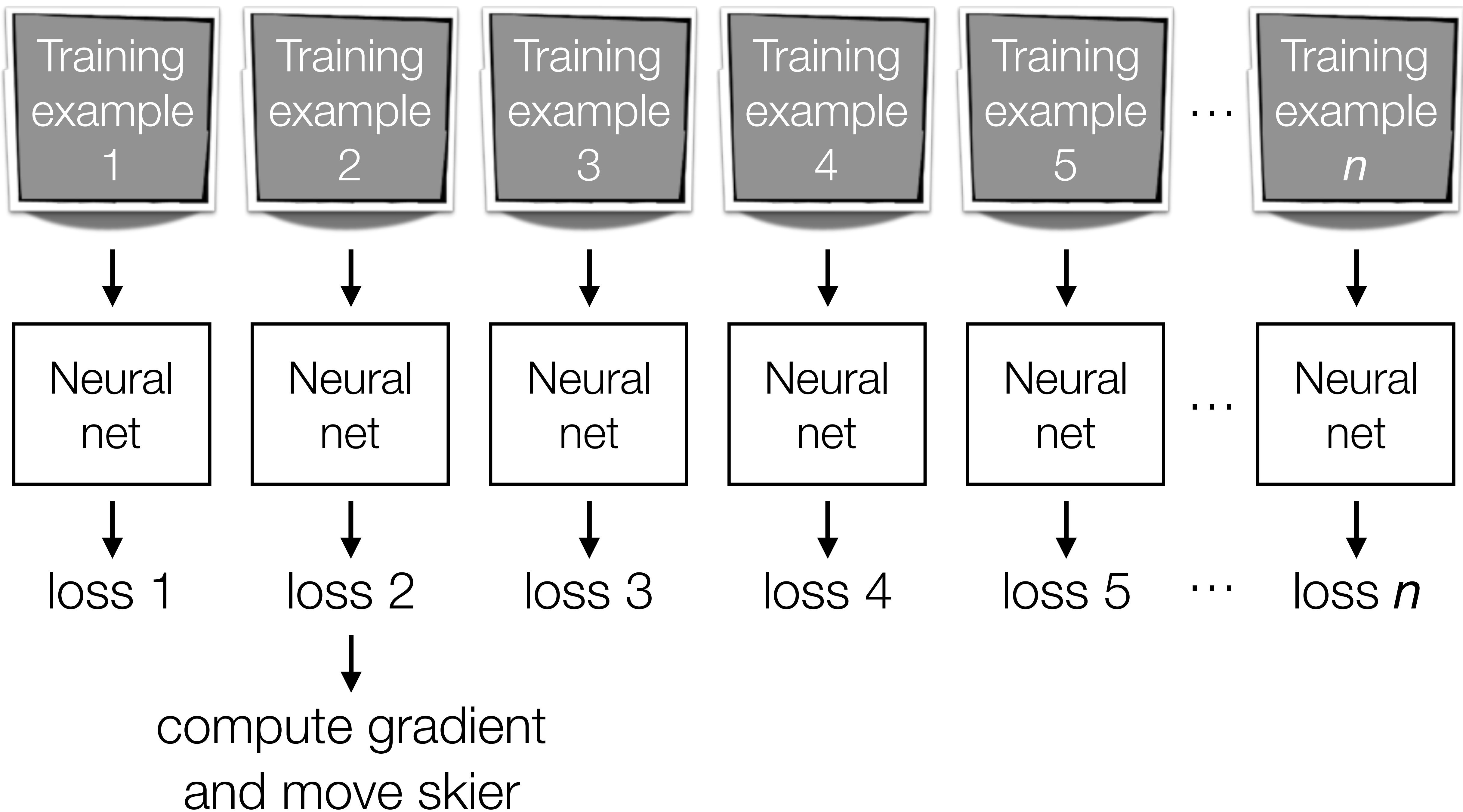
Computing gradients using all the training data seems really expensive!

Stochastic Gradient Descent (SGD)



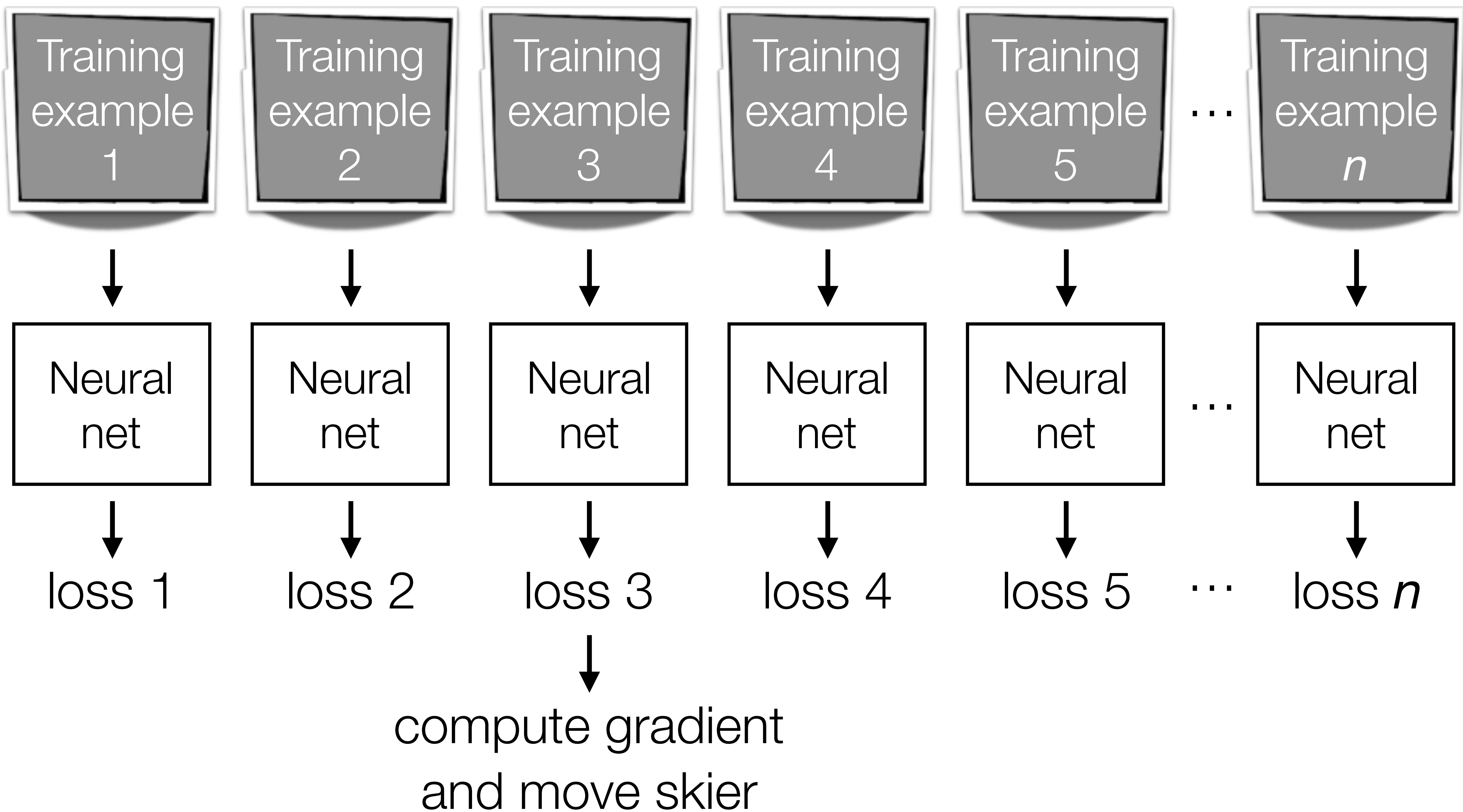
SGD: compute gradient using only 1 training example at a time
(can think of this gradient as a noisy approximation of the “full” gradient)

Stochastic Gradient Descent (SGD)



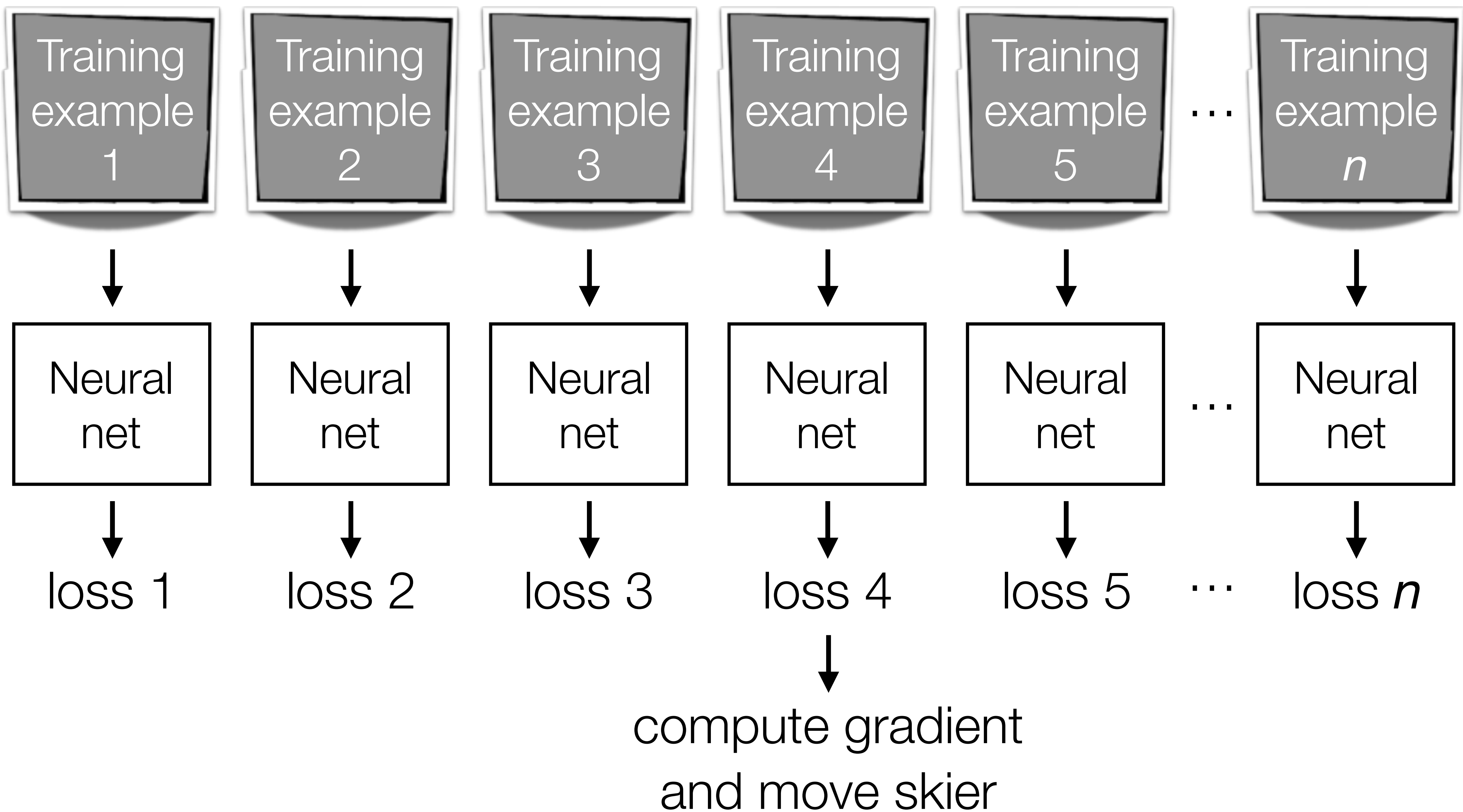
SGD: compute gradient using only 1 training example at a time
(can think of this gradient as a noisy approximation of the "full" gradient)

Stochastic Gradient Descent (SGD)



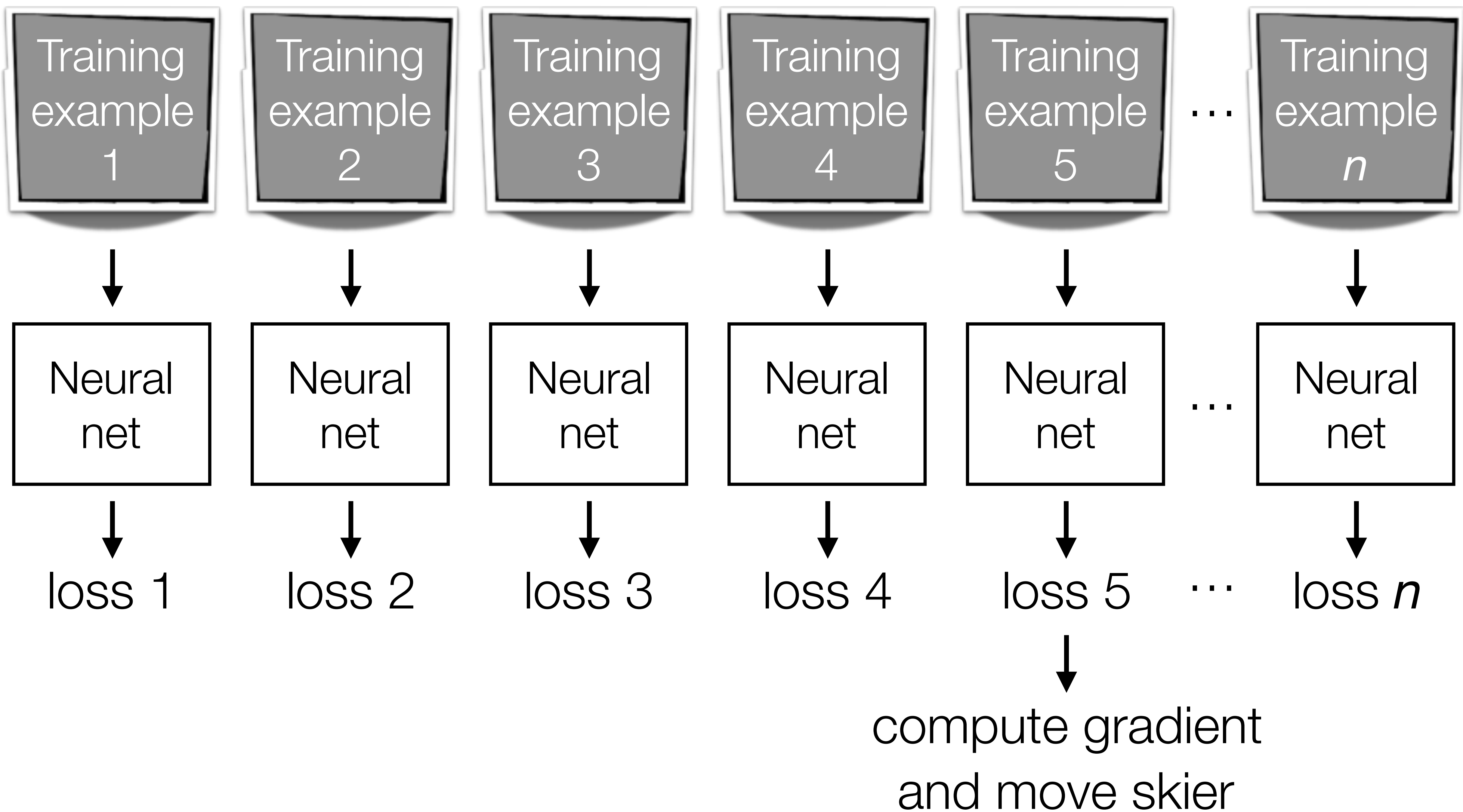
SGD: compute gradient using only 1 training example at a time
(can think of this gradient as a noisy approximation of the "full" gradient)

Stochastic Gradient Descent (SGD)



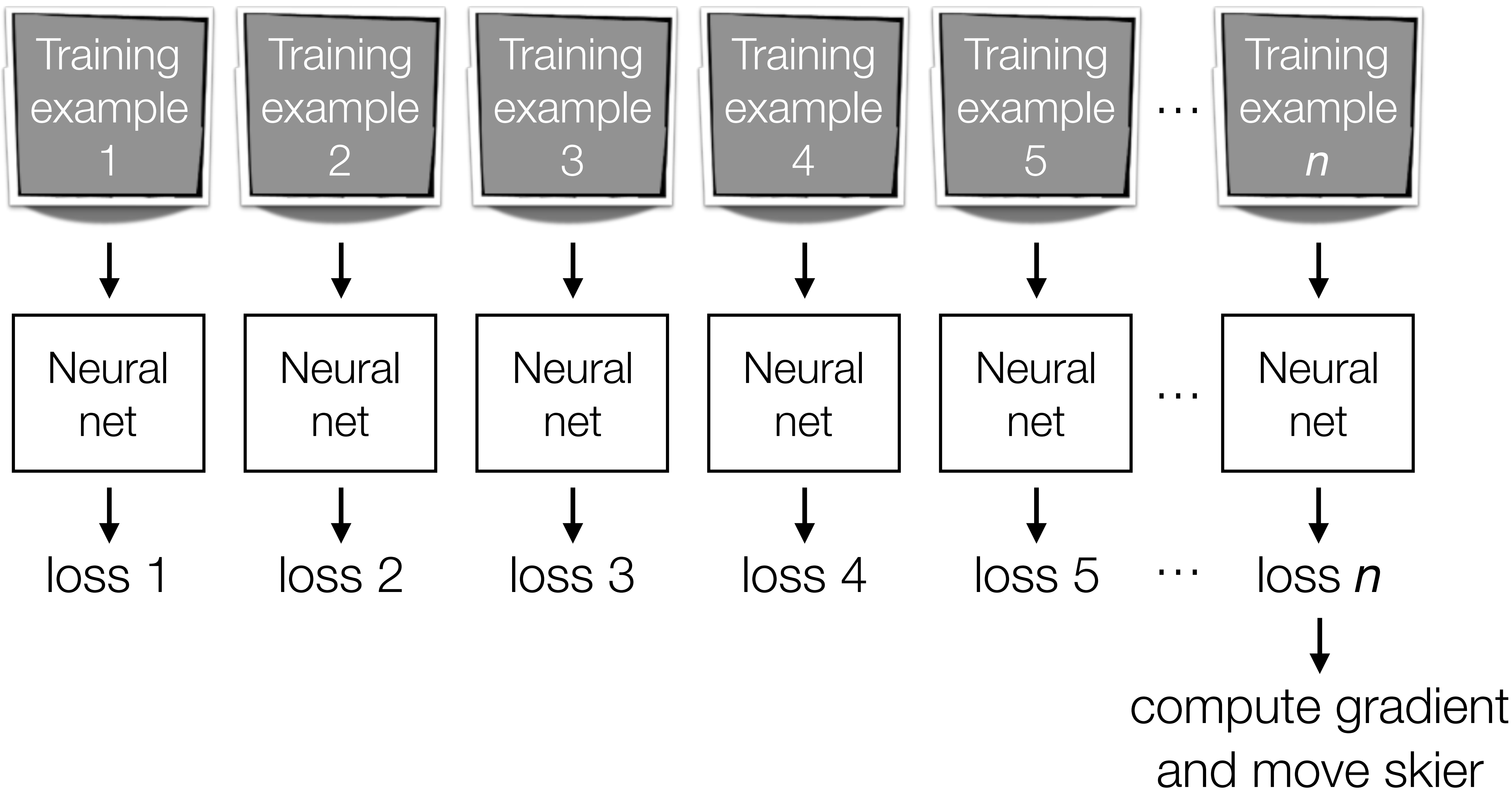
SGD: compute gradient using only 1 training example at a time
(can think of this gradient as a noisy approximation of the "full" gradient)

Stochastic Gradient Descent (SGD)



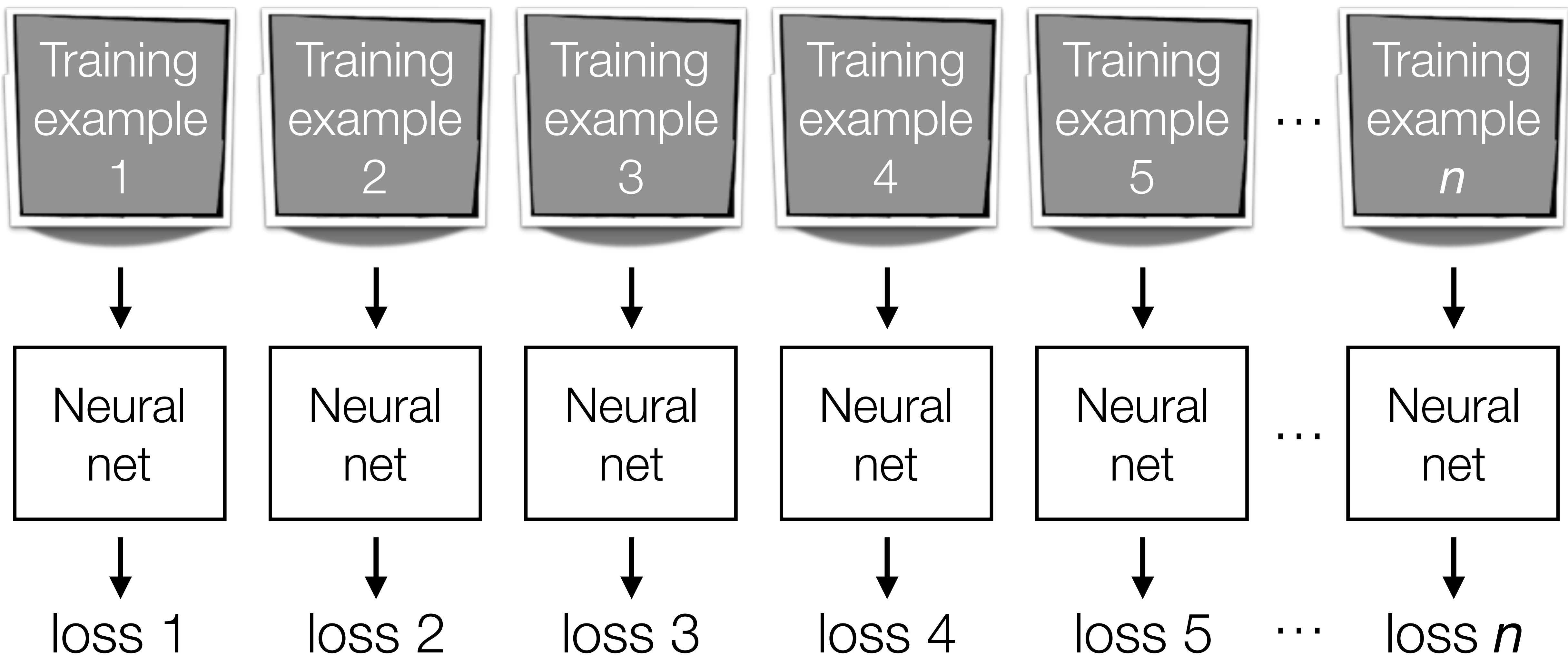
SGD: compute gradient using only 1 training example at a time
(can think of this gradient as a noisy approximation of the "full" gradient)

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)

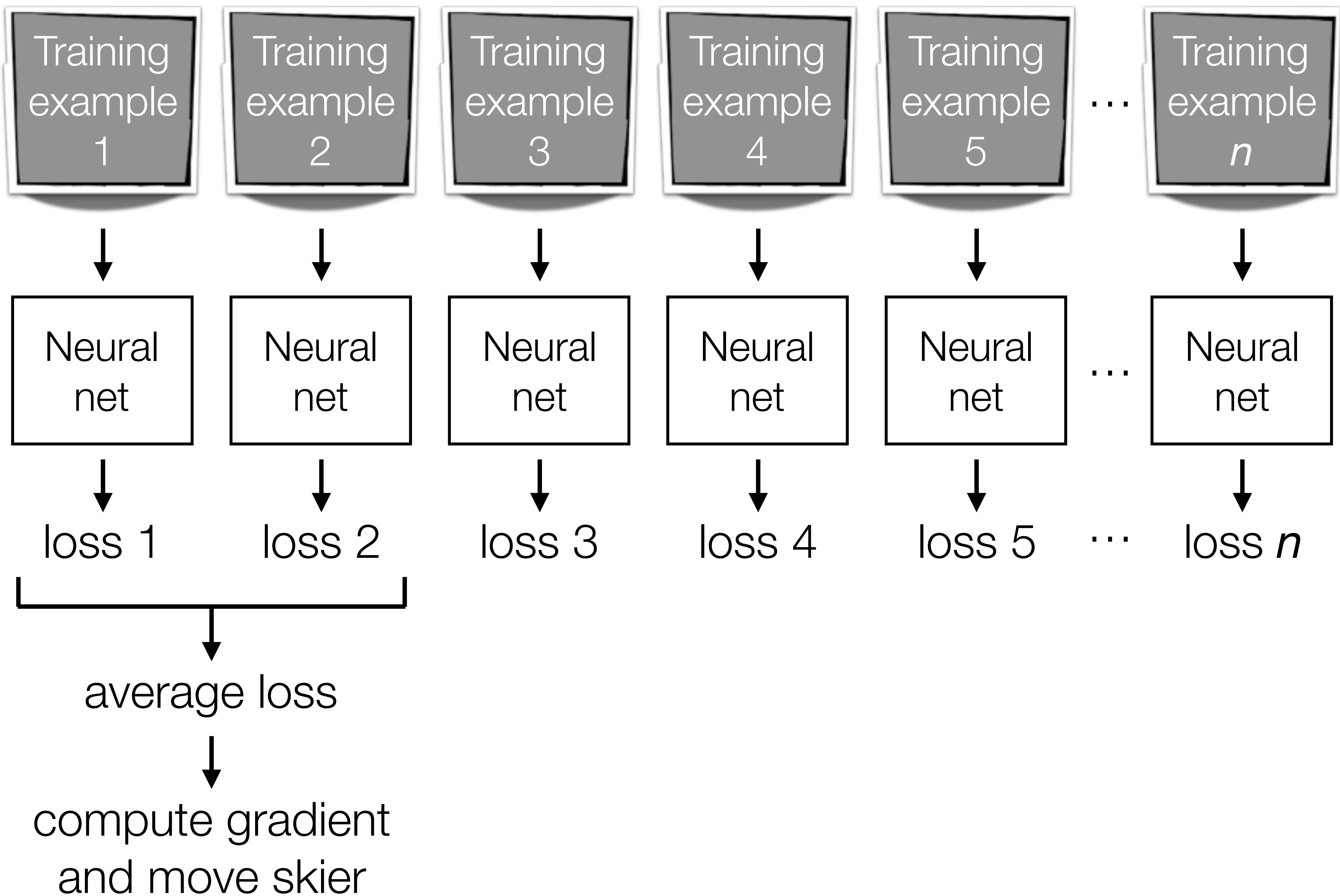


compute gradient
and move skier

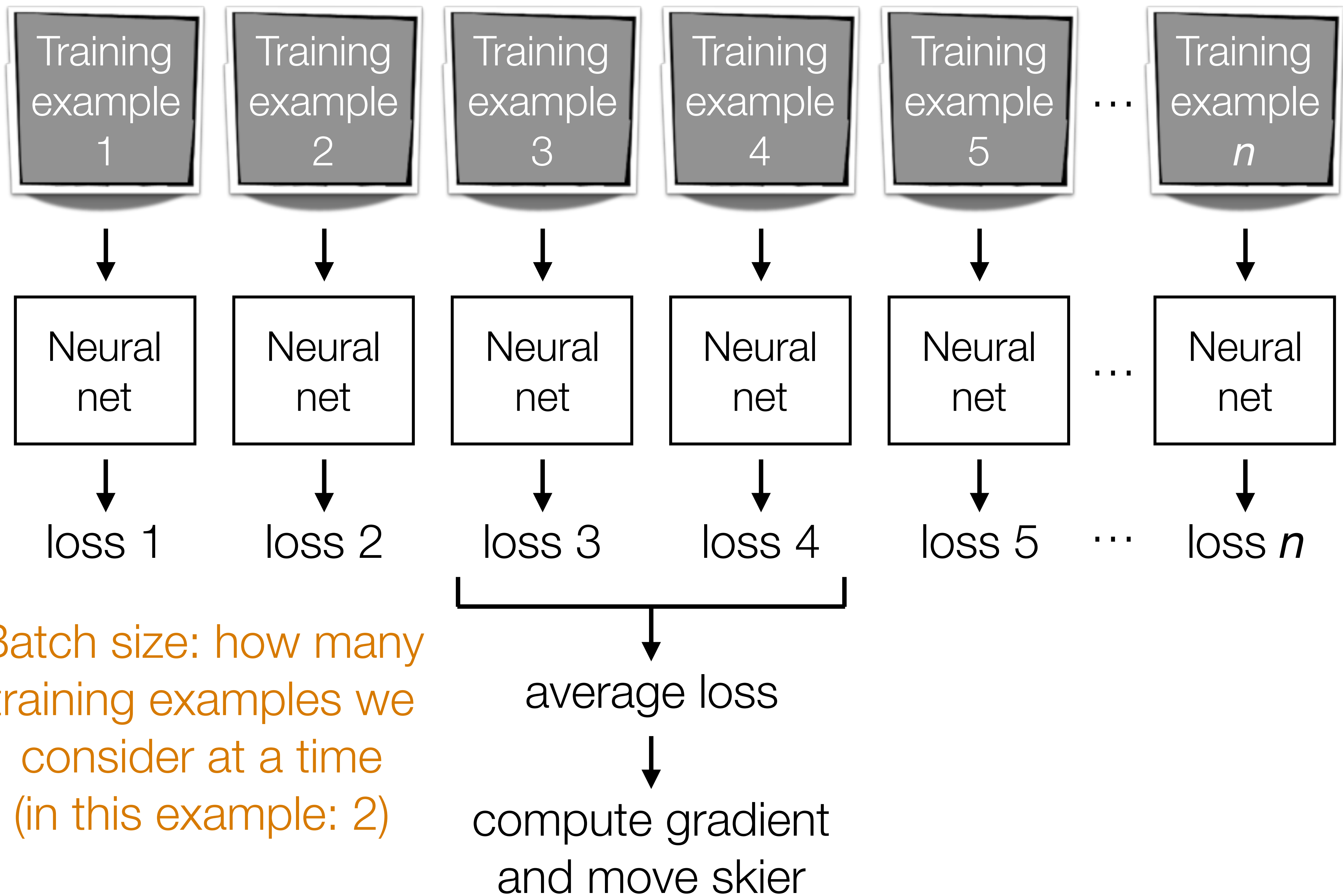
An epoch refers to 1 full pass
through all the training data

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



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

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)

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

Example: classify between Tesla's and Toyota's

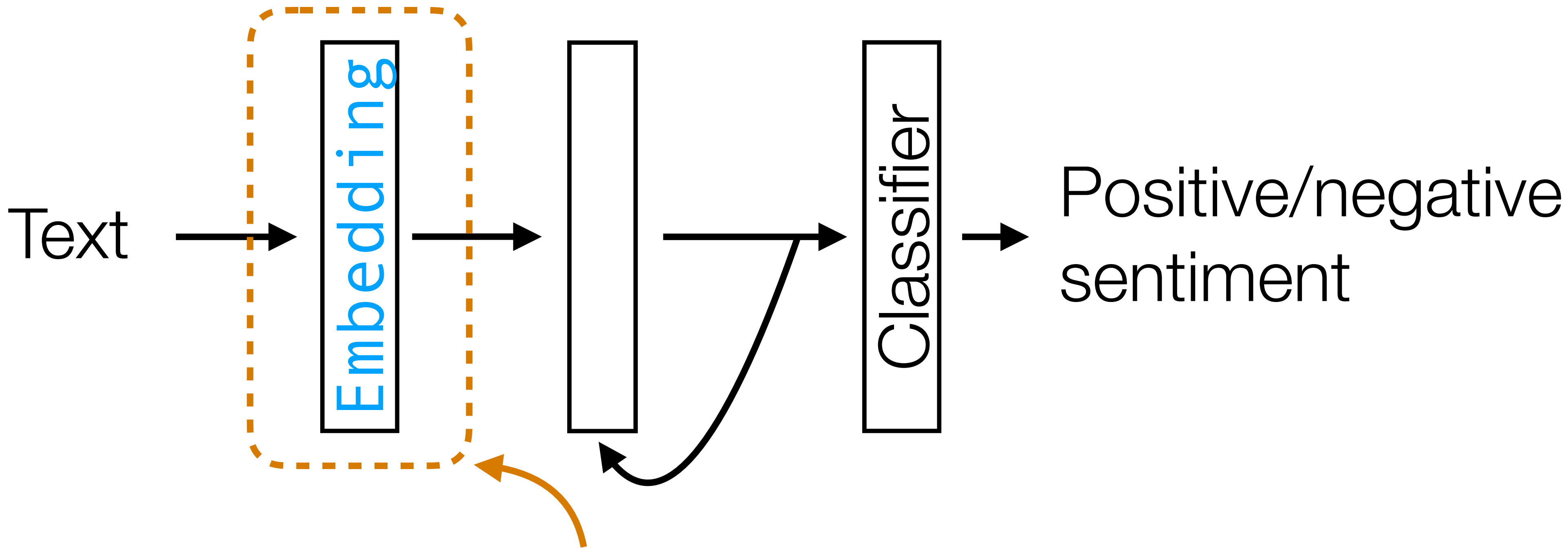
You collect photos from the internet of both, but your dataset size is small, on the order of 1000 images

Strategy: take existing pre-trained CNN for ImageNet classification and change final layer to do classification between Tesla's and Toyota's rather than classifying into 1000 objects

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

Example: sentiment analysis RNN demo



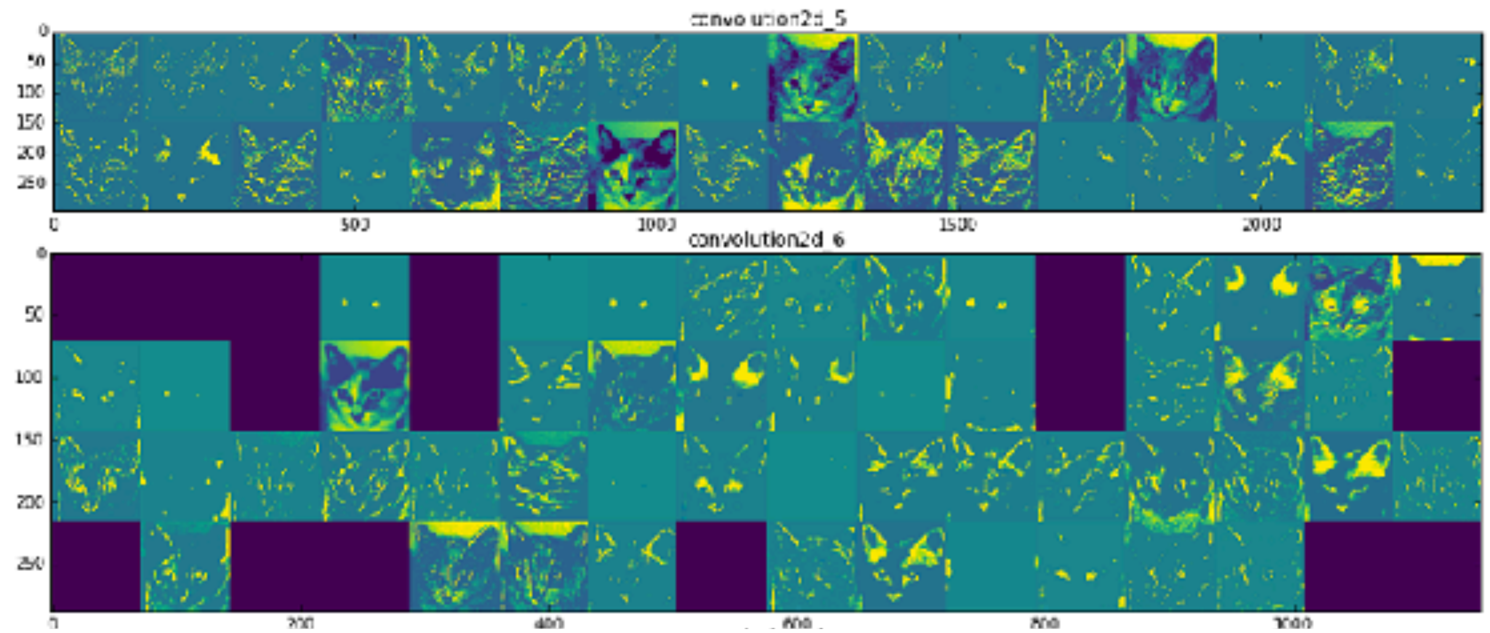
We fixed the weights here to come from GloVe and disabled training for this layer!

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

IMDb review dataset is small in comparison

Visualizing What a Deep Net Learned

- Very straight-forward for CNNs
  - Plot filter outputs at different layers



  - Plot regions that maximally activate an output neuron

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

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!

Training data point: opioid

"Training label": epidemic, or, crisis, is

There are "positive" examples of what context words are for "opioid"

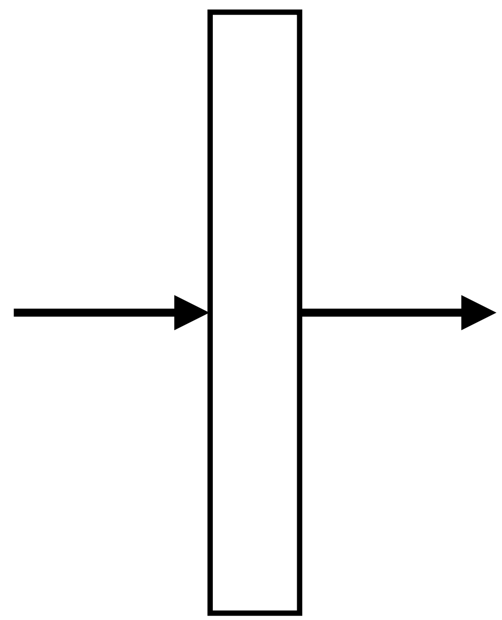
Also provide "negative" examples of words that are *not* likely to be context words (e.g., randomly sample words elsewhere in document)

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!

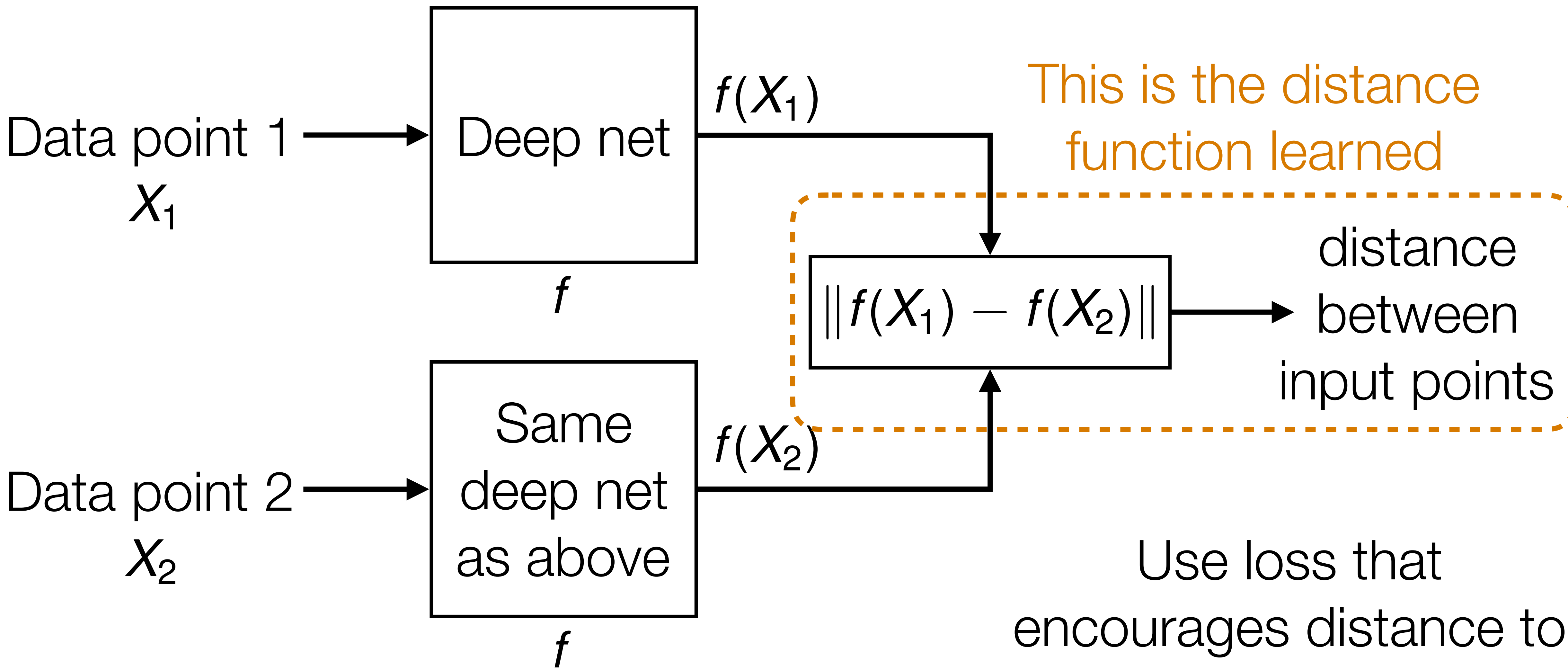
- Key idea: predict part of the training data from other parts of the training data

- 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



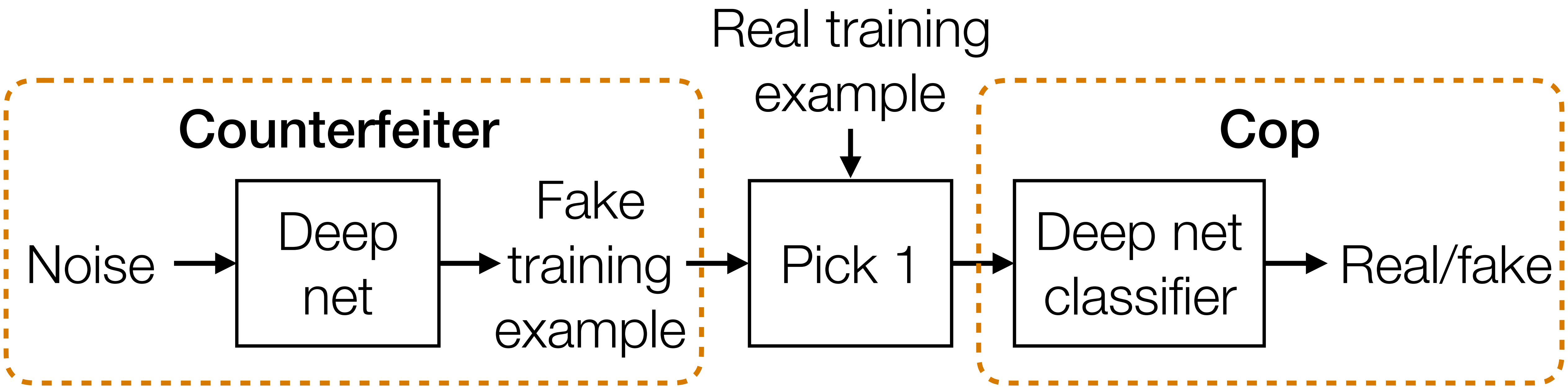
Note: we are learning the function f

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

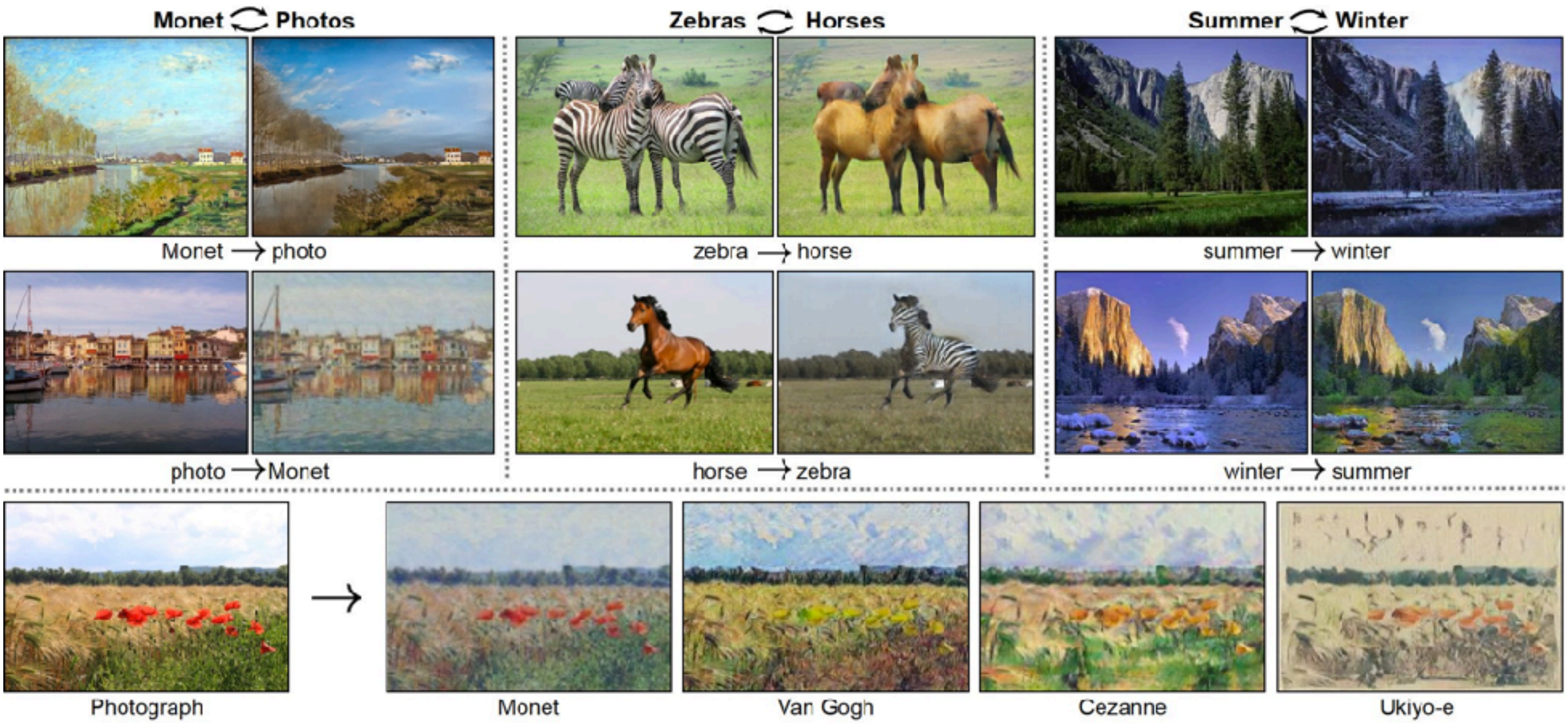
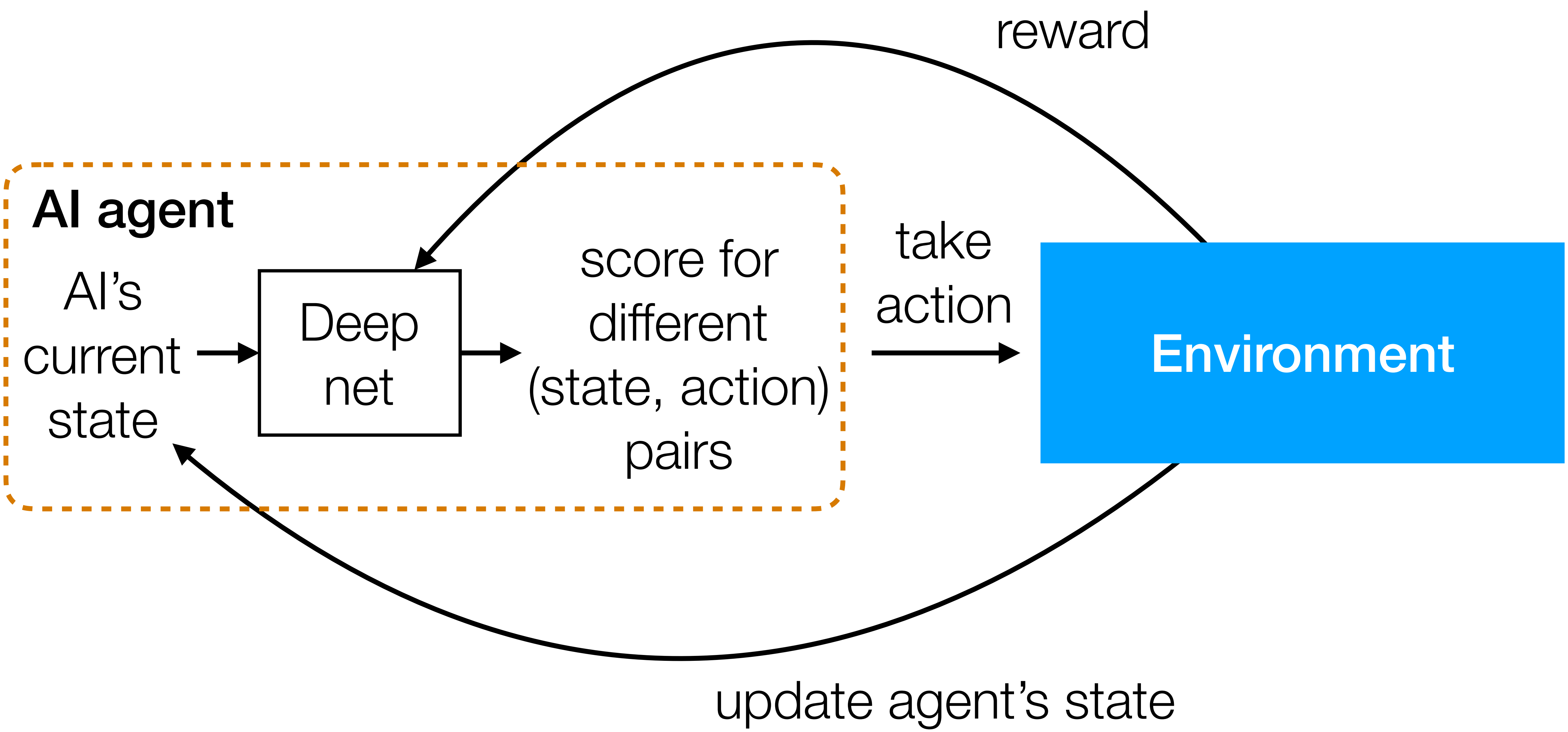


Image-to-image translation results from UC Berkeley using GANs
(Isola et al 2017, Zhu et al 2017)

Deep Reinforcement Learning

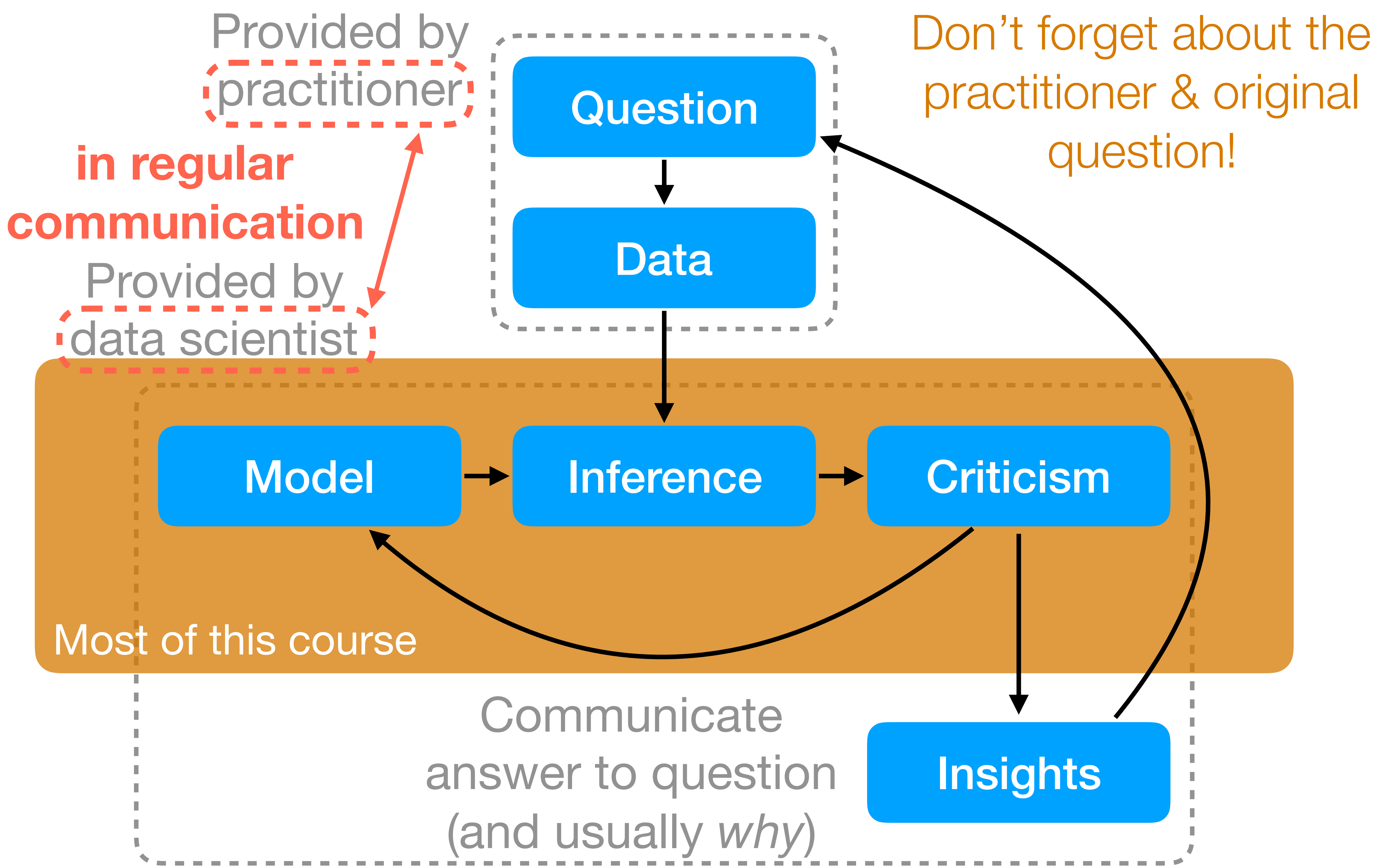
The machinery behind AlphaGo and similar systems



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?

95-865



Provided by
practitioner
in regular
communication
Provided by
data scientist
Question
Data
Don’t forget about the
practitioner & original
question!
Model
Inference
Criticism
Most of this course
Communicate
answer to question
(and usually *why*)
Insights

95-865 Some Parting Thoughts

- Remember to **visualize different steps of your data analysis pipeline** — very helpful when you're still debugging
- Very often there are *tons* of models that you could 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

Thanks for being a beta tester!