

95-865 Lecture 13:
Learning a Deep Net, Other
Deep Learning Topics, Wrap-up

George Chen

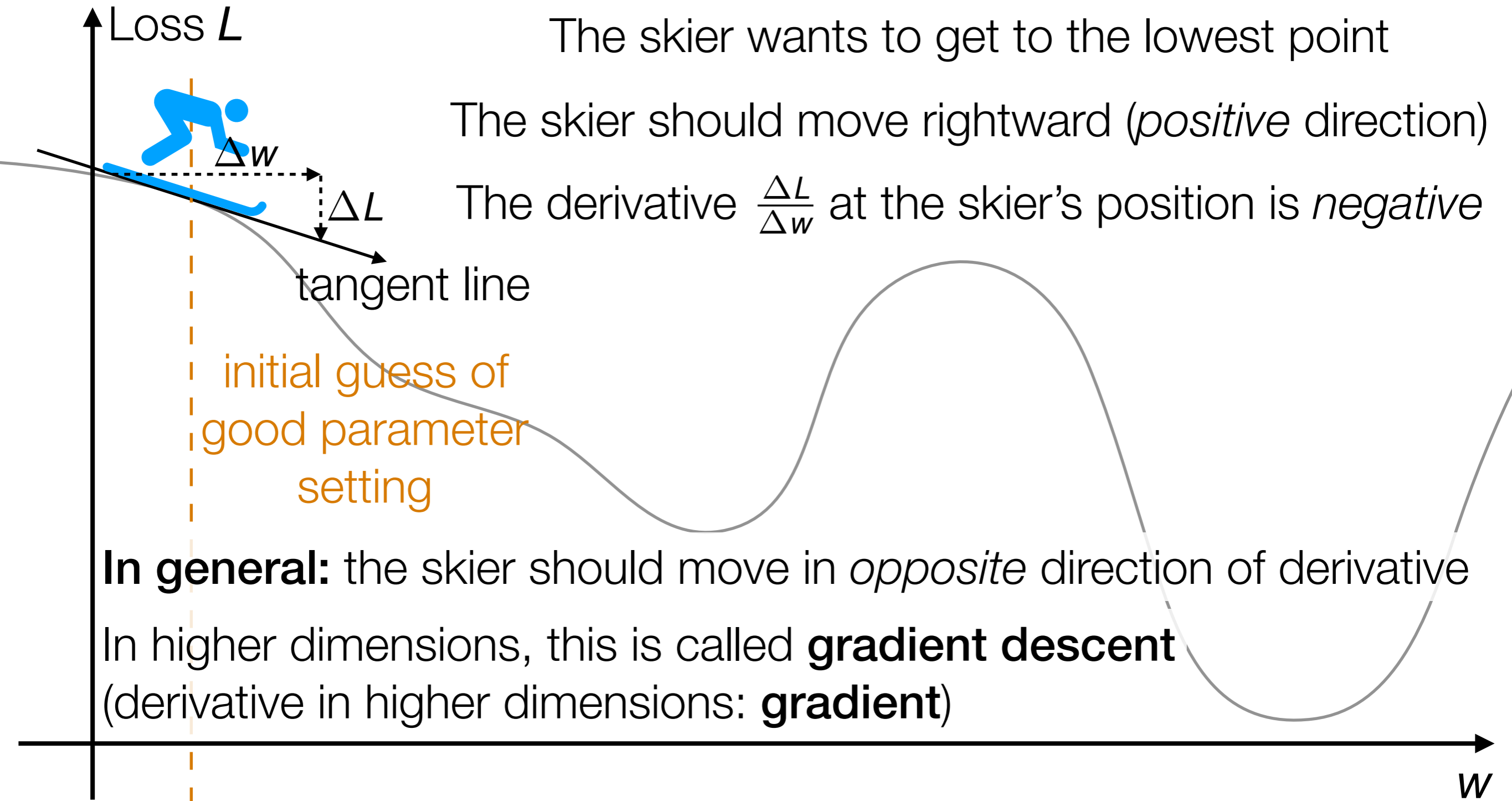
Today

- How learning a deep net roughly works
- High-level overview of a bunch of deep learning topics we didn't get to
- Course wrap-up

Learning a Deep Net

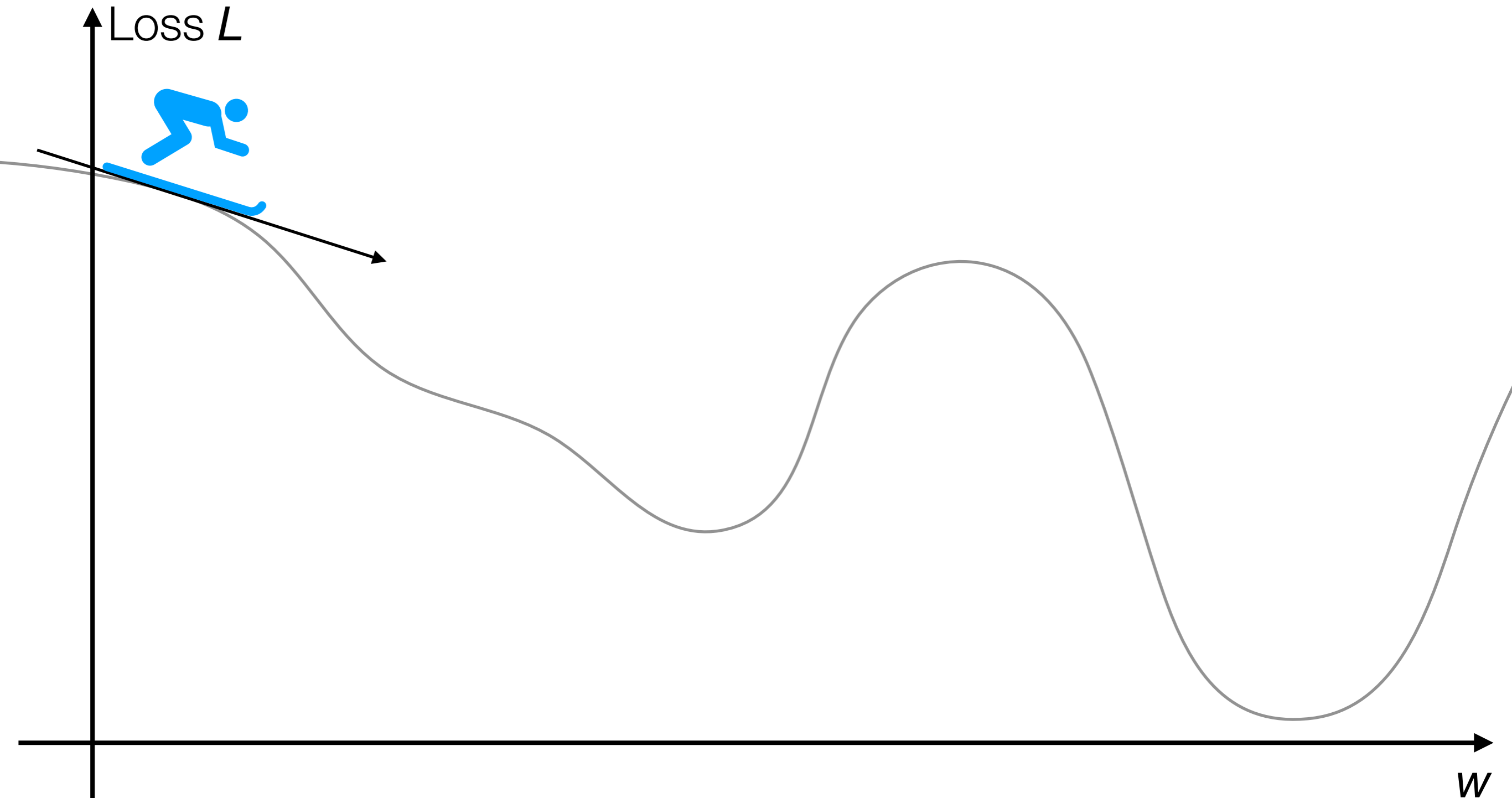
Gradient Descent

Suppose the neural network has a single real number parameter w



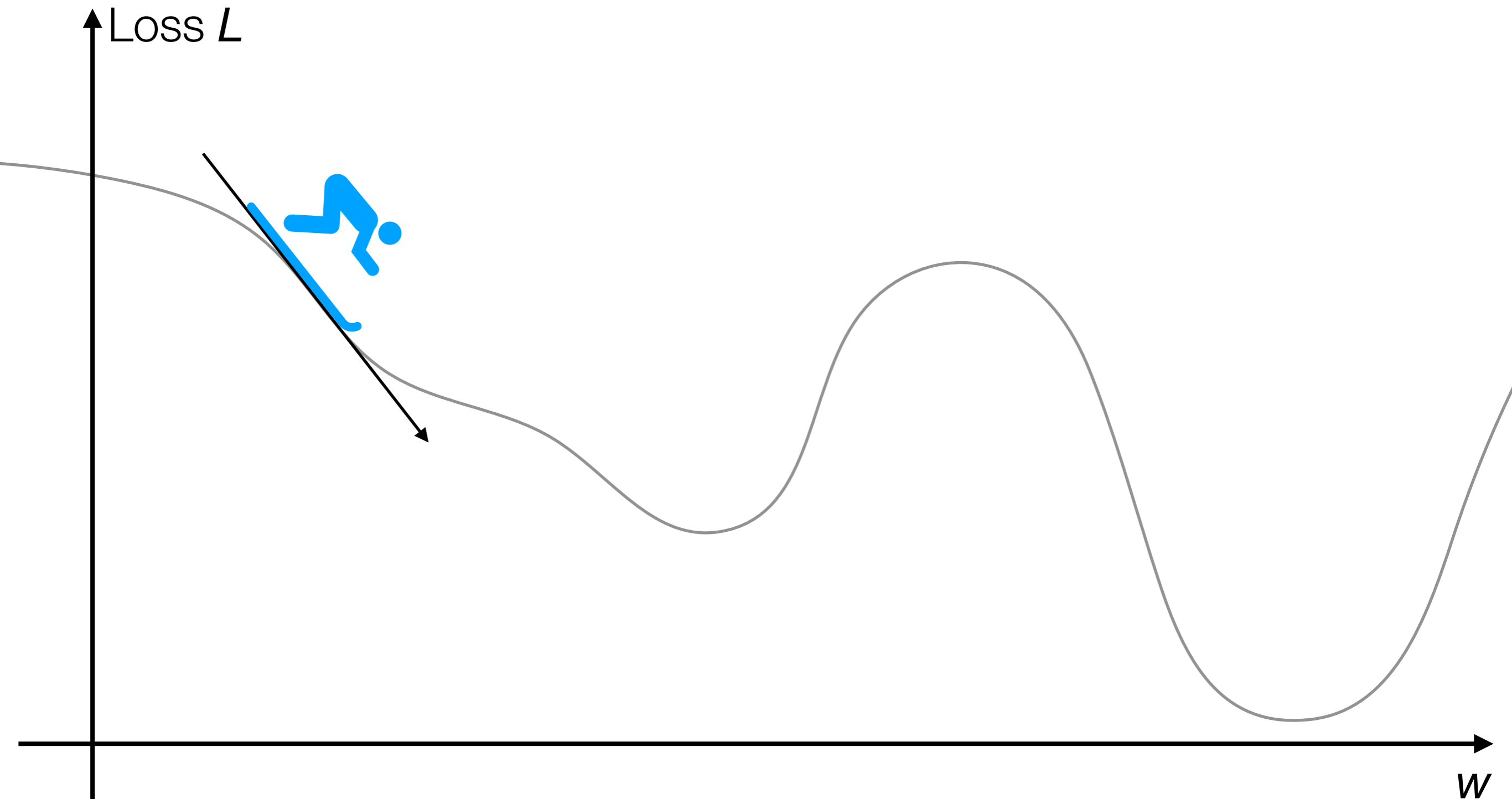
Gradient Descent

Suppose the neural network has a single real number parameter w



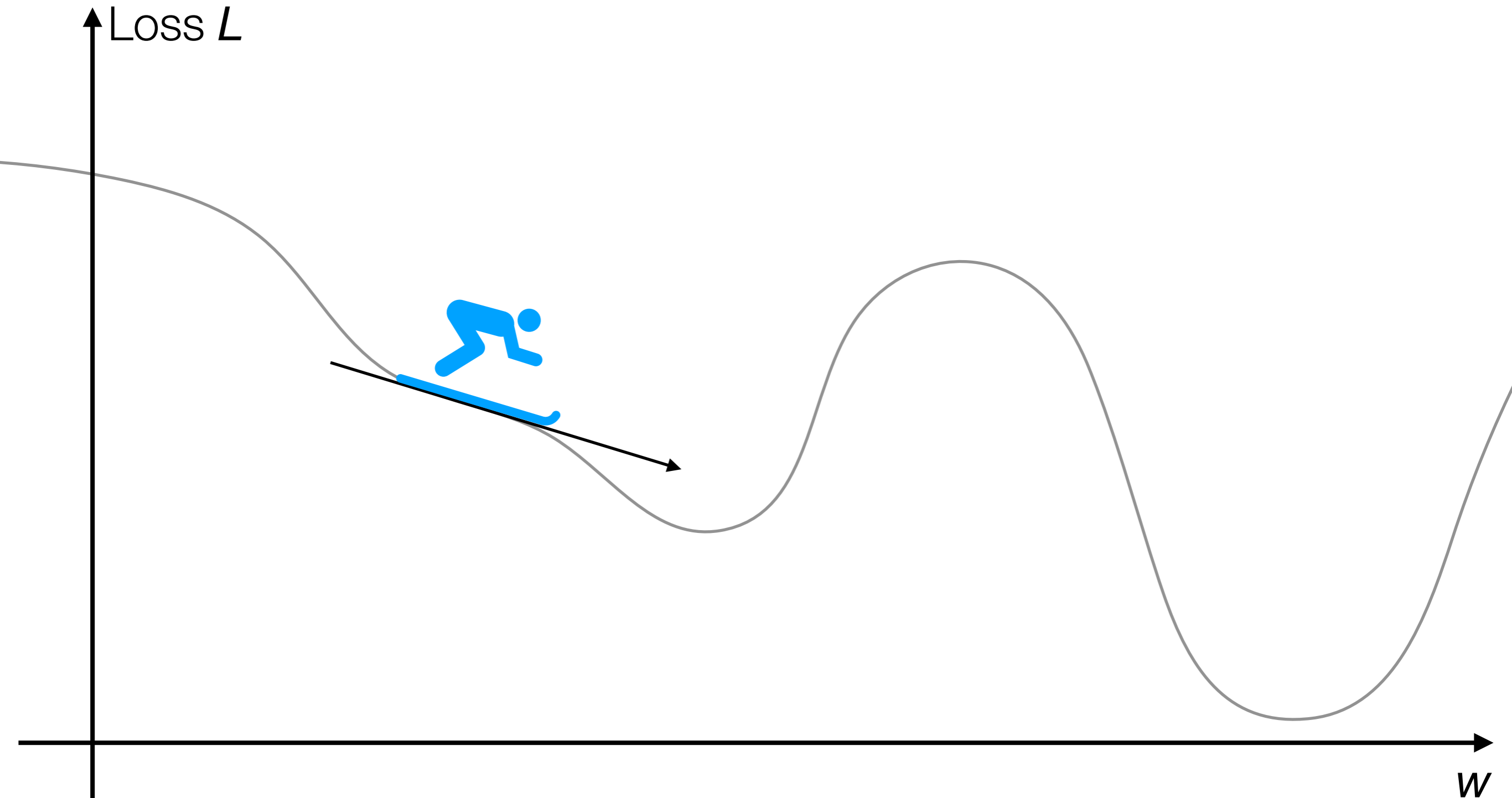
Gradient Descent

Suppose the neural network has a single real number parameter w



Gradient Descent

Suppose the neural network has a single real number parameter w

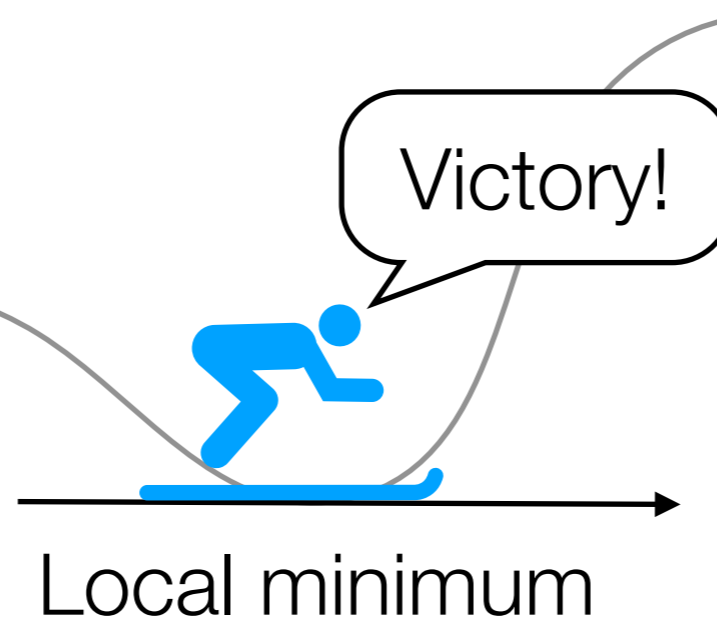


Gradient Descent

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

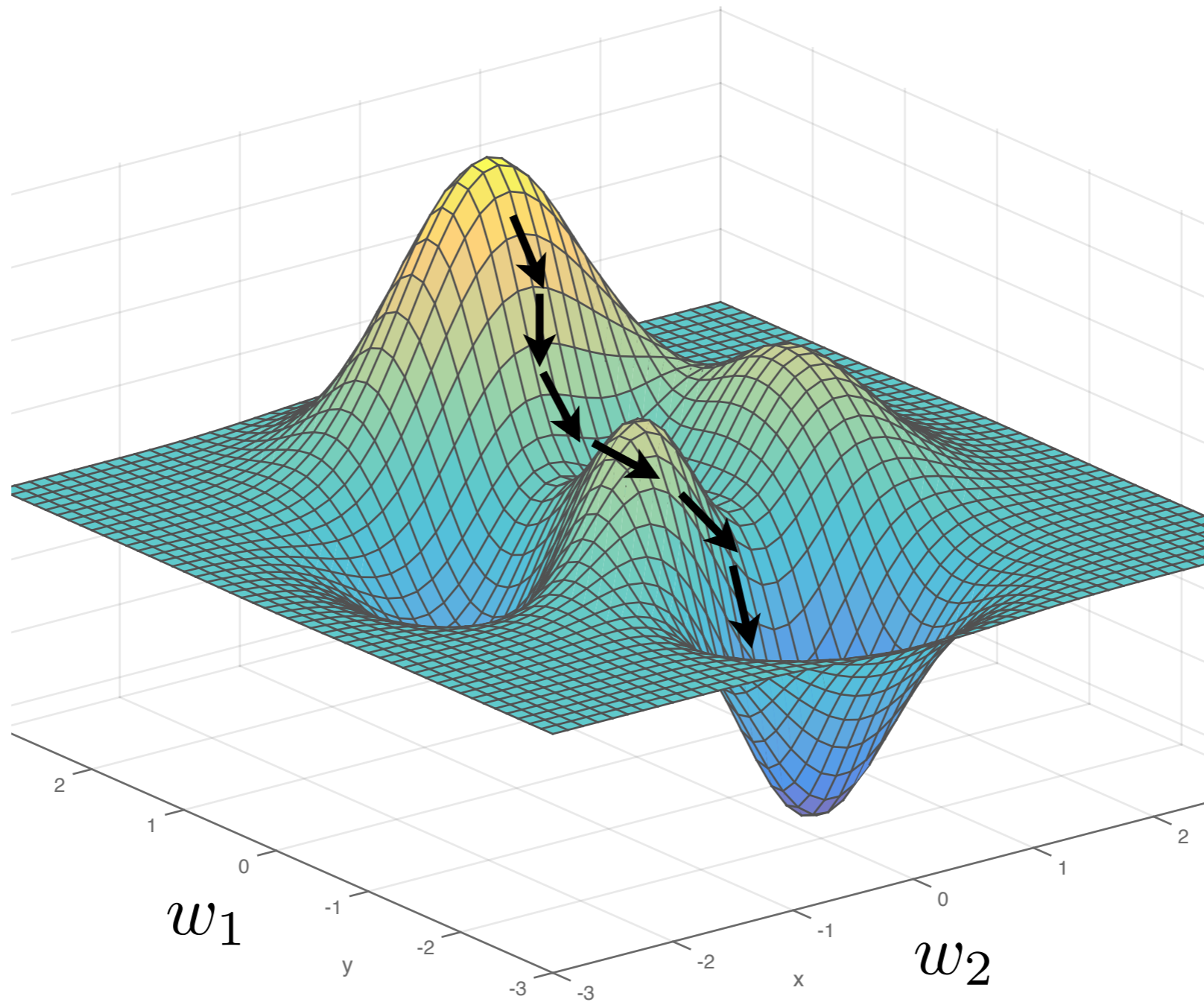


In practice: local minimum often good enough

Gradient Descent

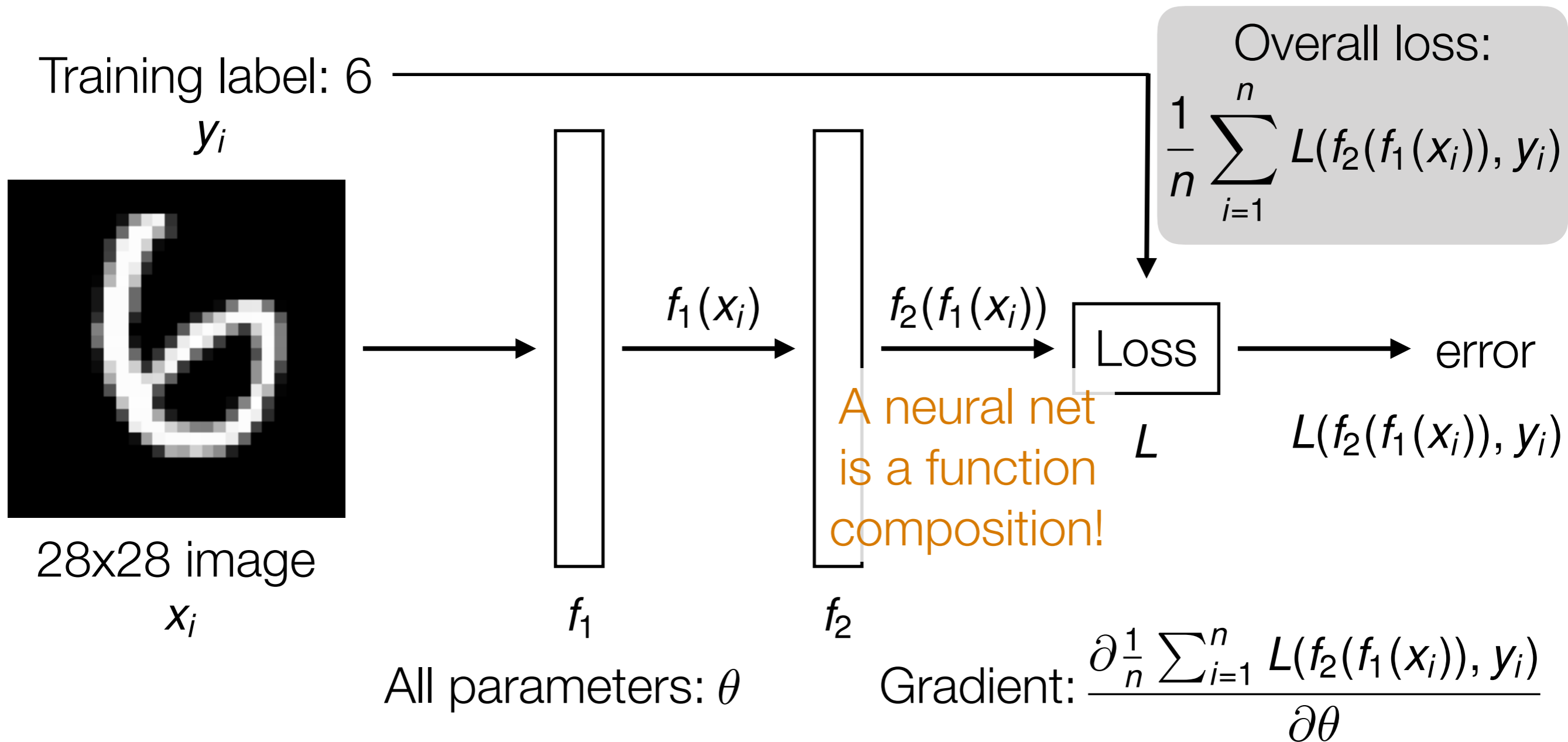
2D example

$L(\mathbf{w})$



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

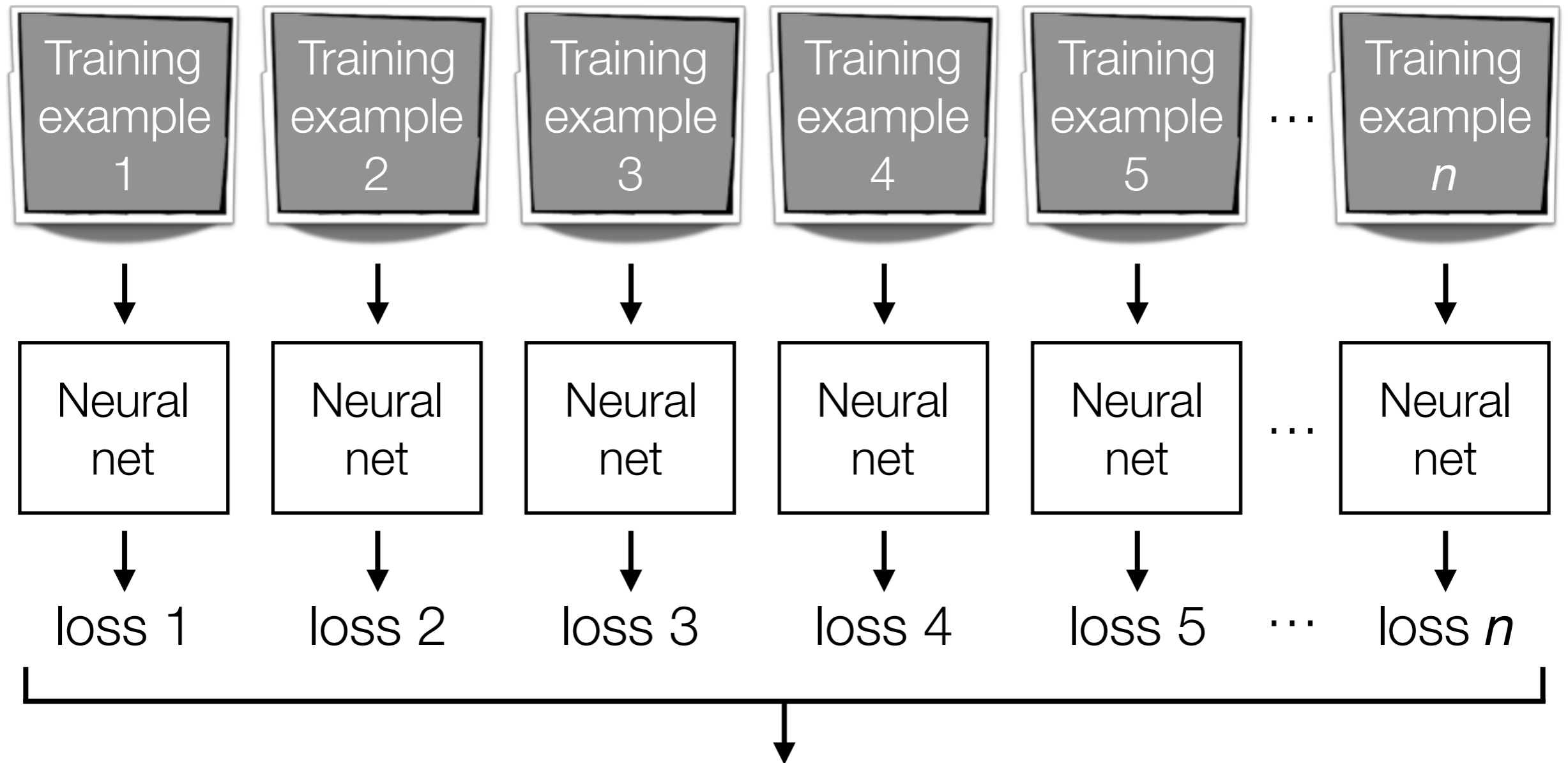
Handwritten Digit Recognition



Automatic differentiation is crucial in learning deep nets!

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

Gradient Descent

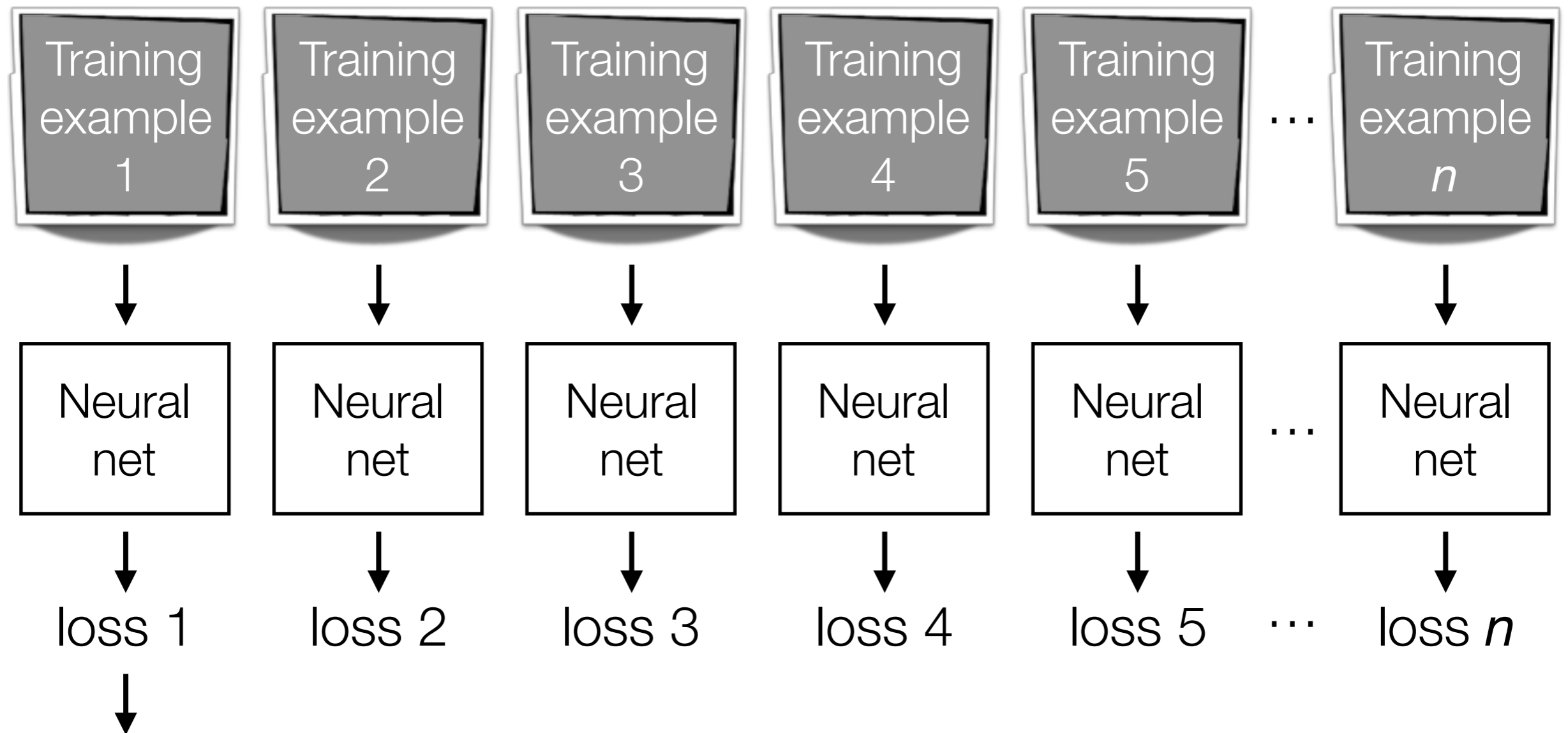


We have to compute lots of gradients to help the skier know where to go!

average loss
↓
compute gradient and move skier

Computing gradients using all the training data seems really expensive!

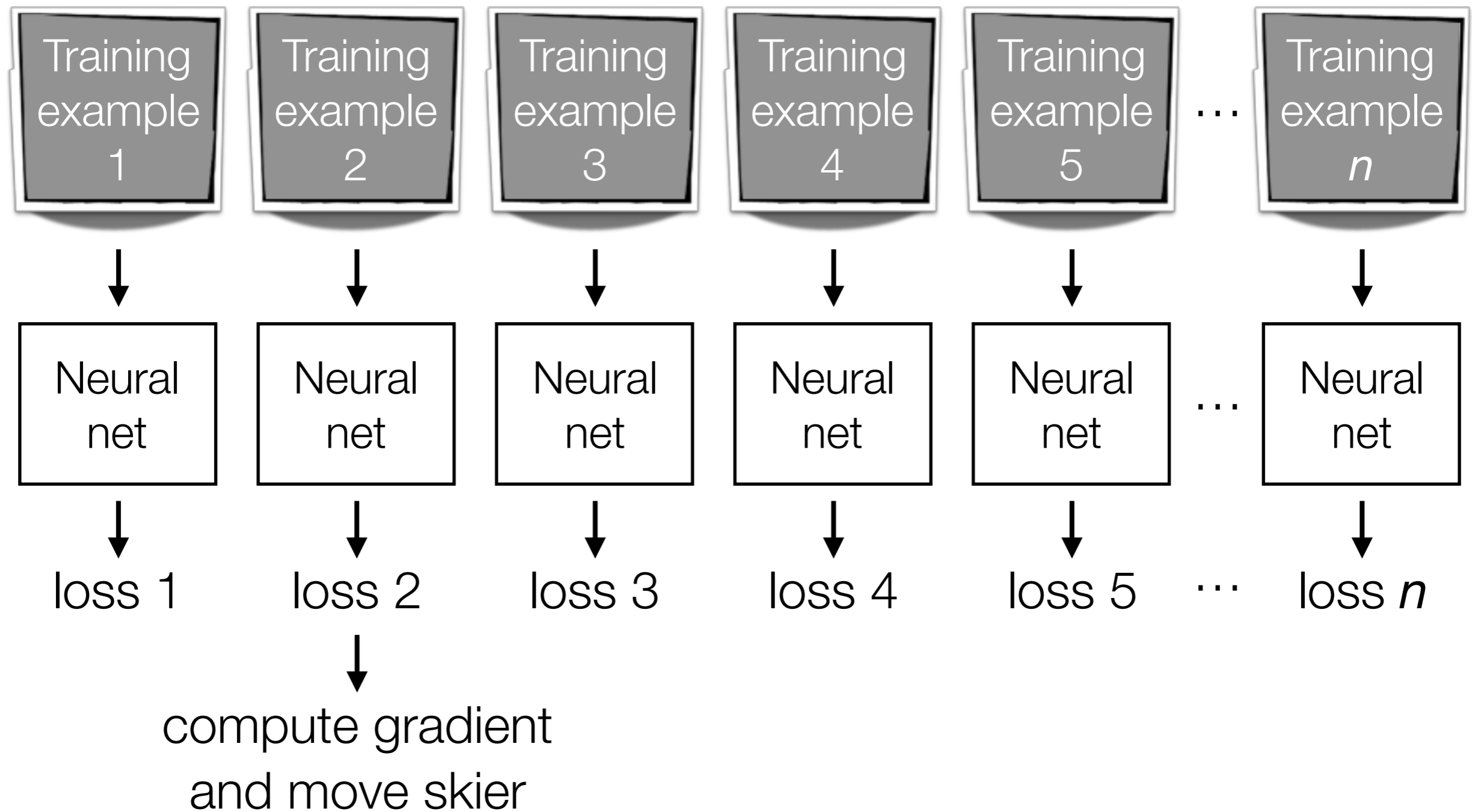
Stochastic Gradient Descent (SGD)



compute gradient
and move skier

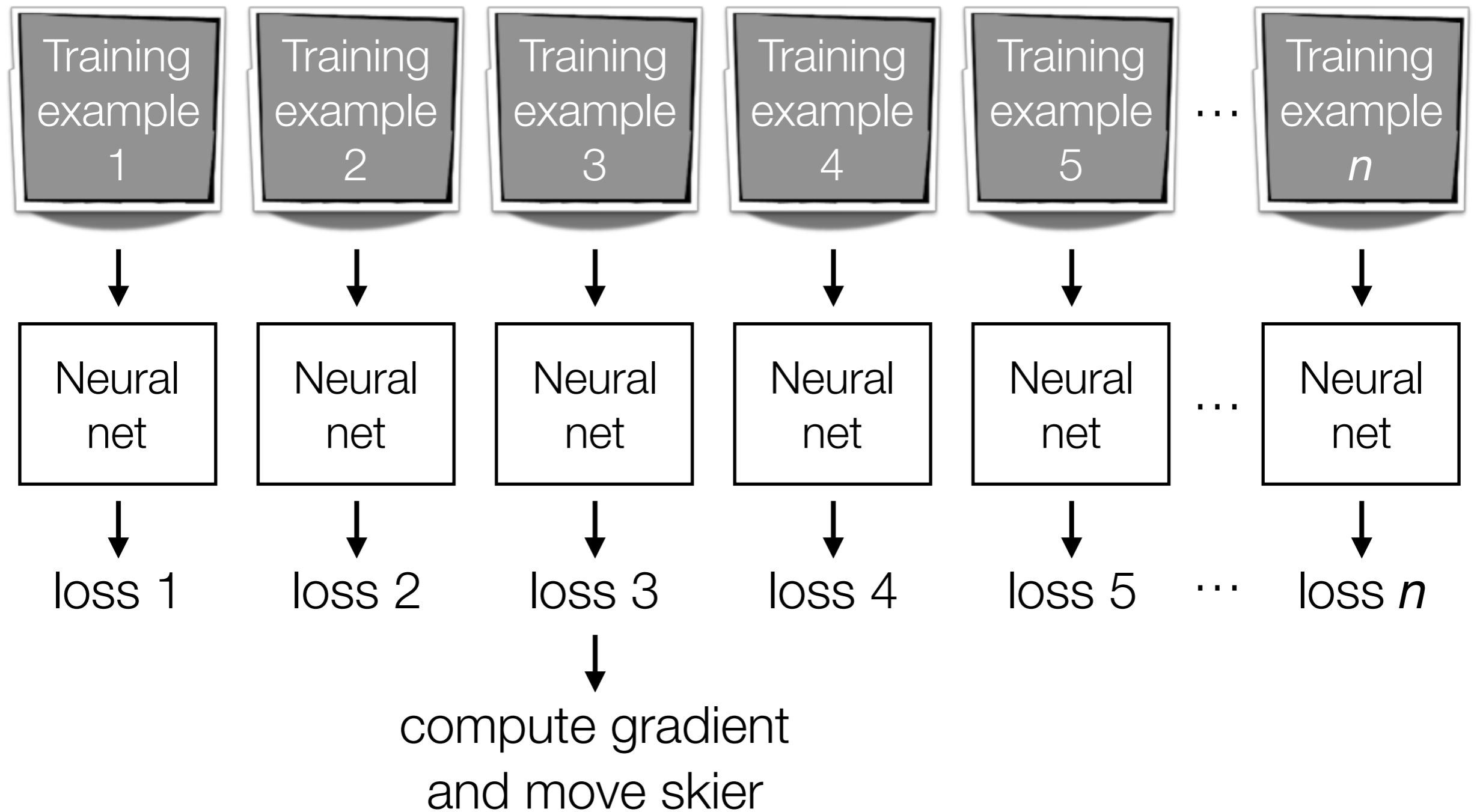
SGD: compute gradient using only 1 training example at a time
(can think of this gradient as a noisy approximation of the “full” gradient)

Stochastic Gradient Descent (SGD)



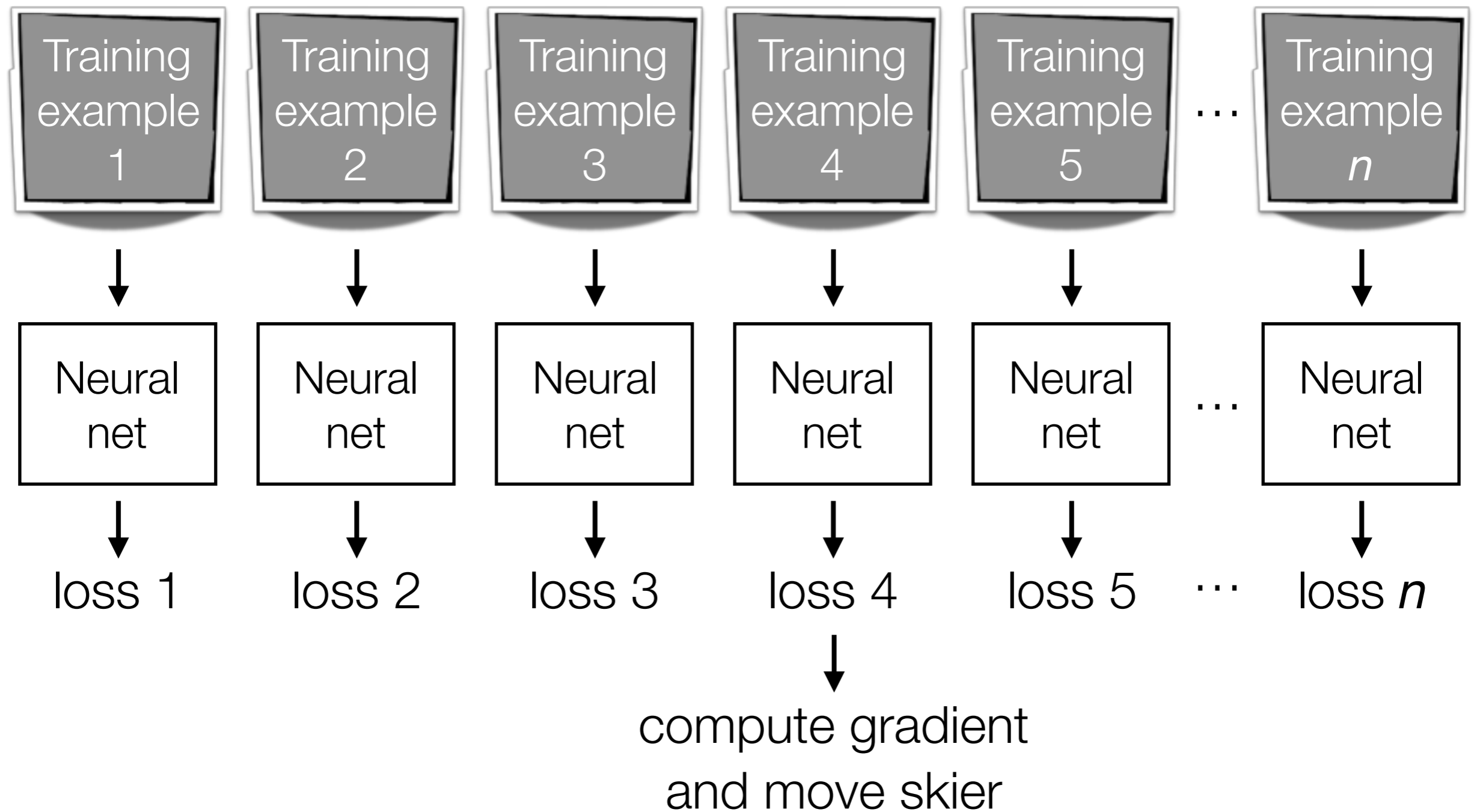
SGD: compute gradient using only 1 training example at a time
(can think of this gradient as a noisy approximation of the “full” gradient)

Stochastic Gradient Descent (SGD)



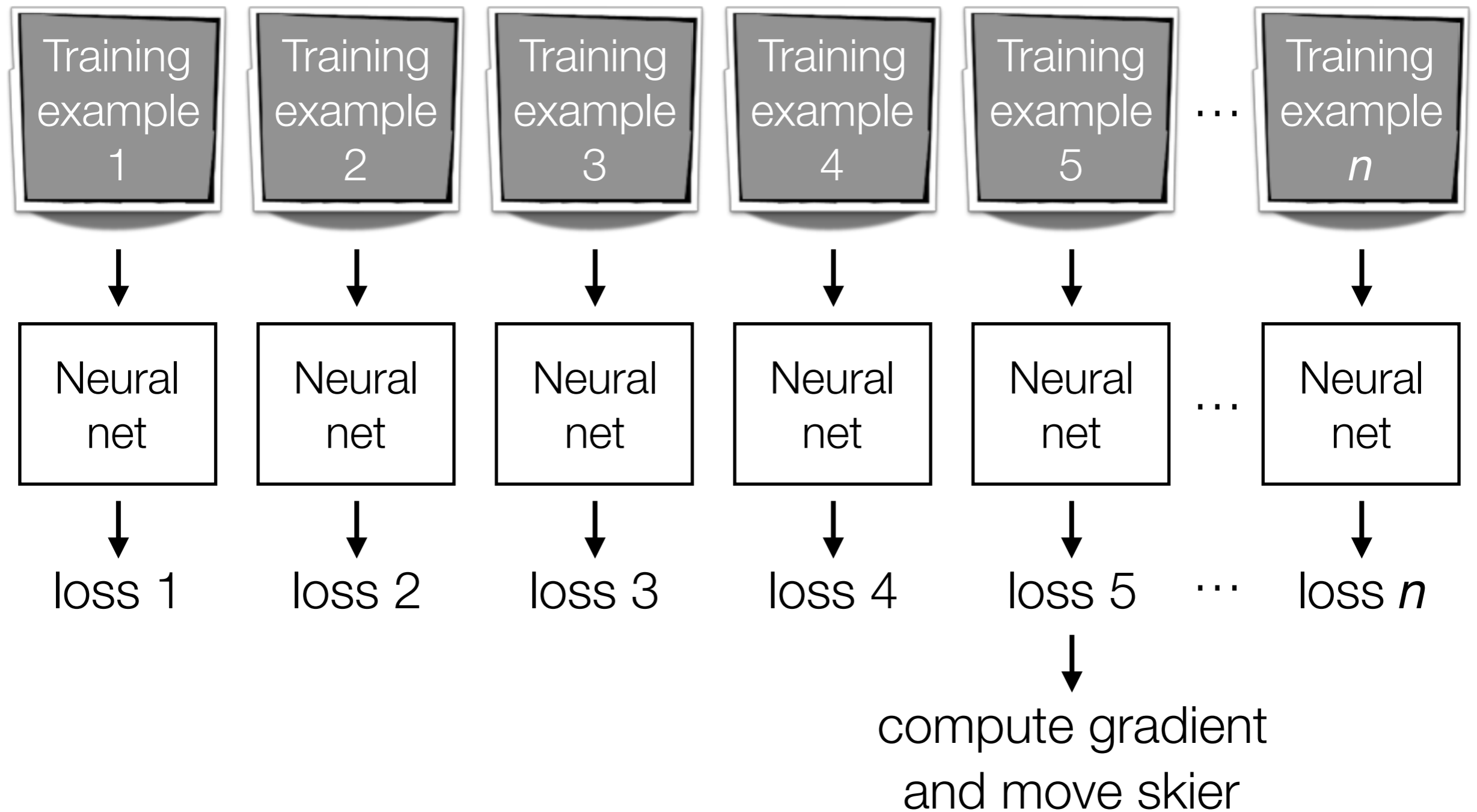
SGD: compute gradient using only 1 training example at a time
(can think of this gradient as a noisy approximation of the “full” gradient)

Stochastic Gradient Descent (SGD)



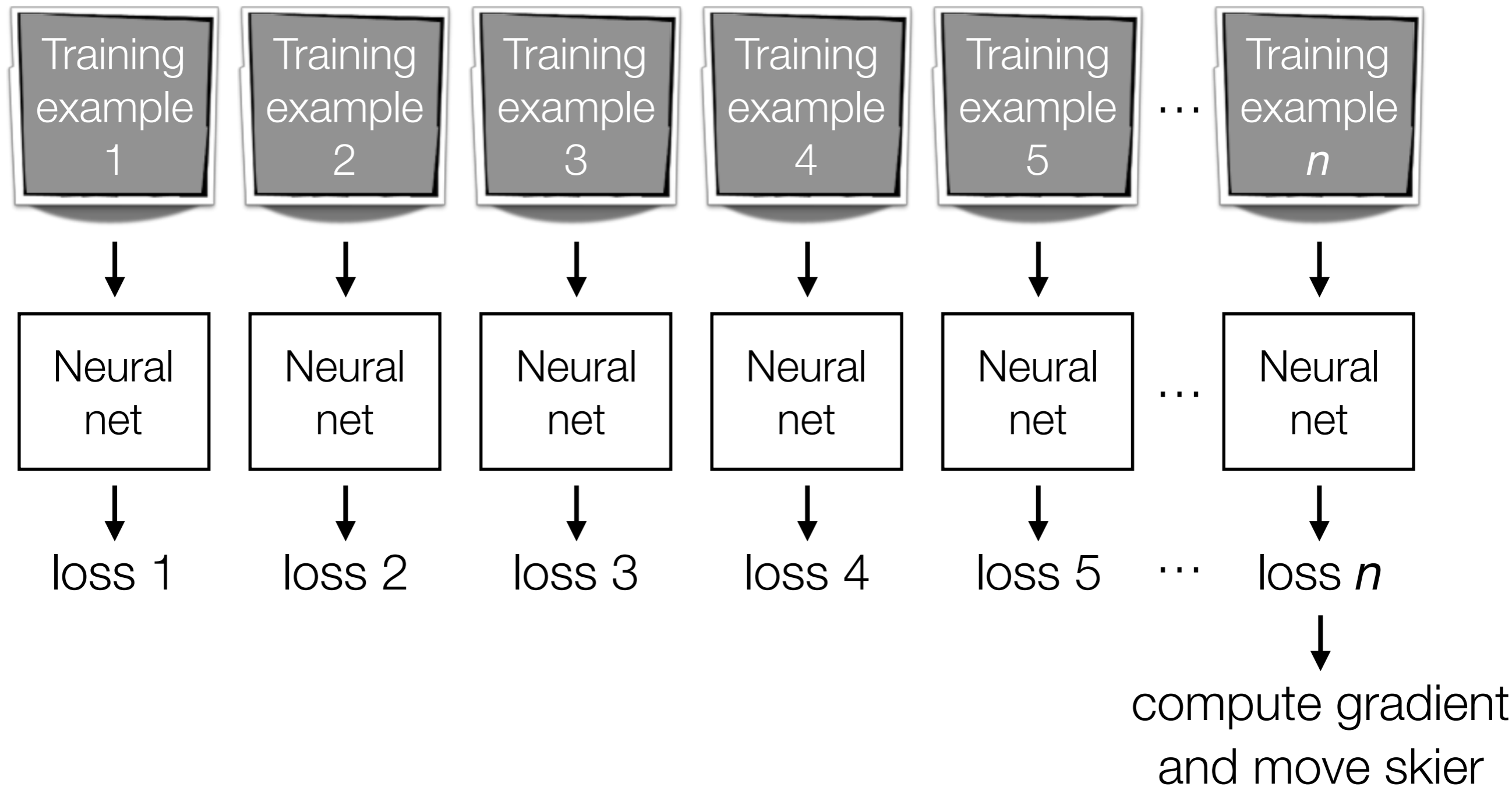
SGD: compute gradient using only 1 training example at a time
(can think of this gradient as a noisy approximation of the “full” gradient)

Stochastic Gradient Descent (SGD)



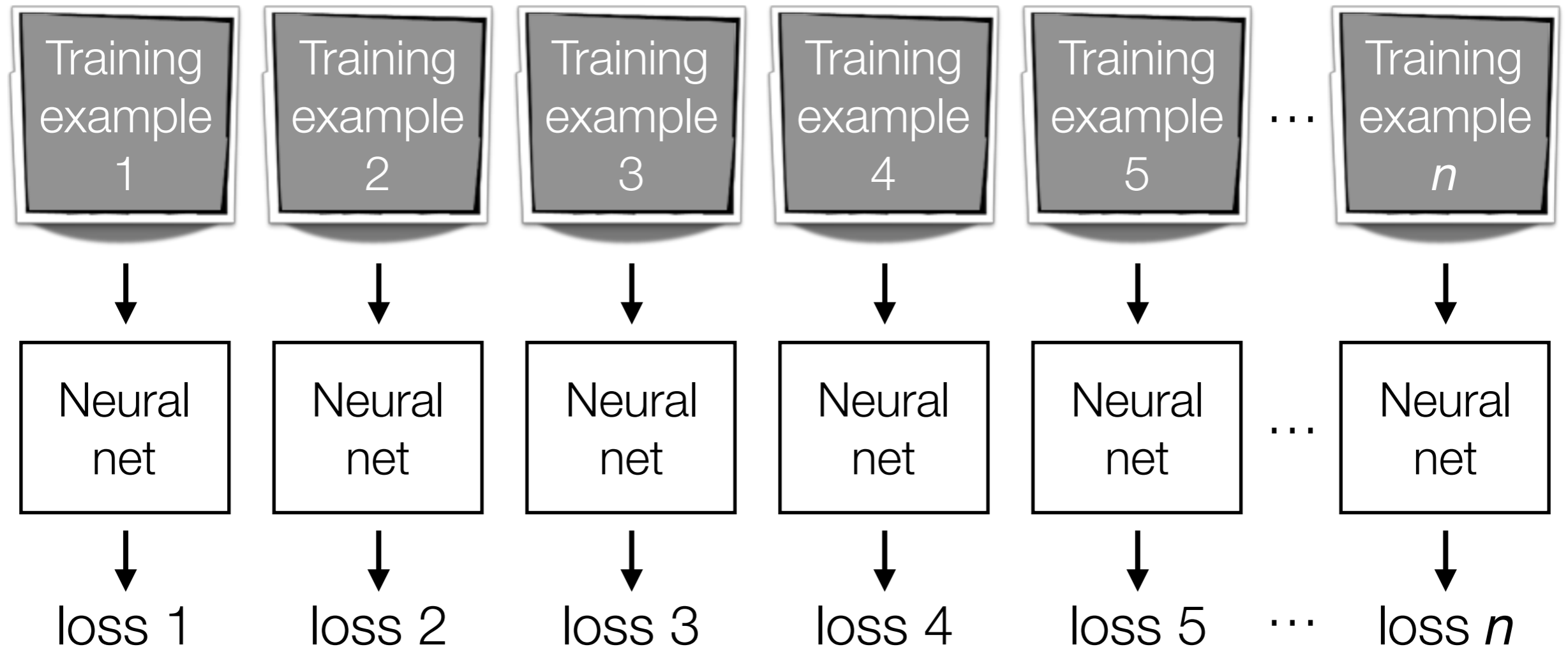
SGD: compute gradient using only 1 training example at a time
(can think of this gradient as a noisy approximation of the “full” gradient)

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)

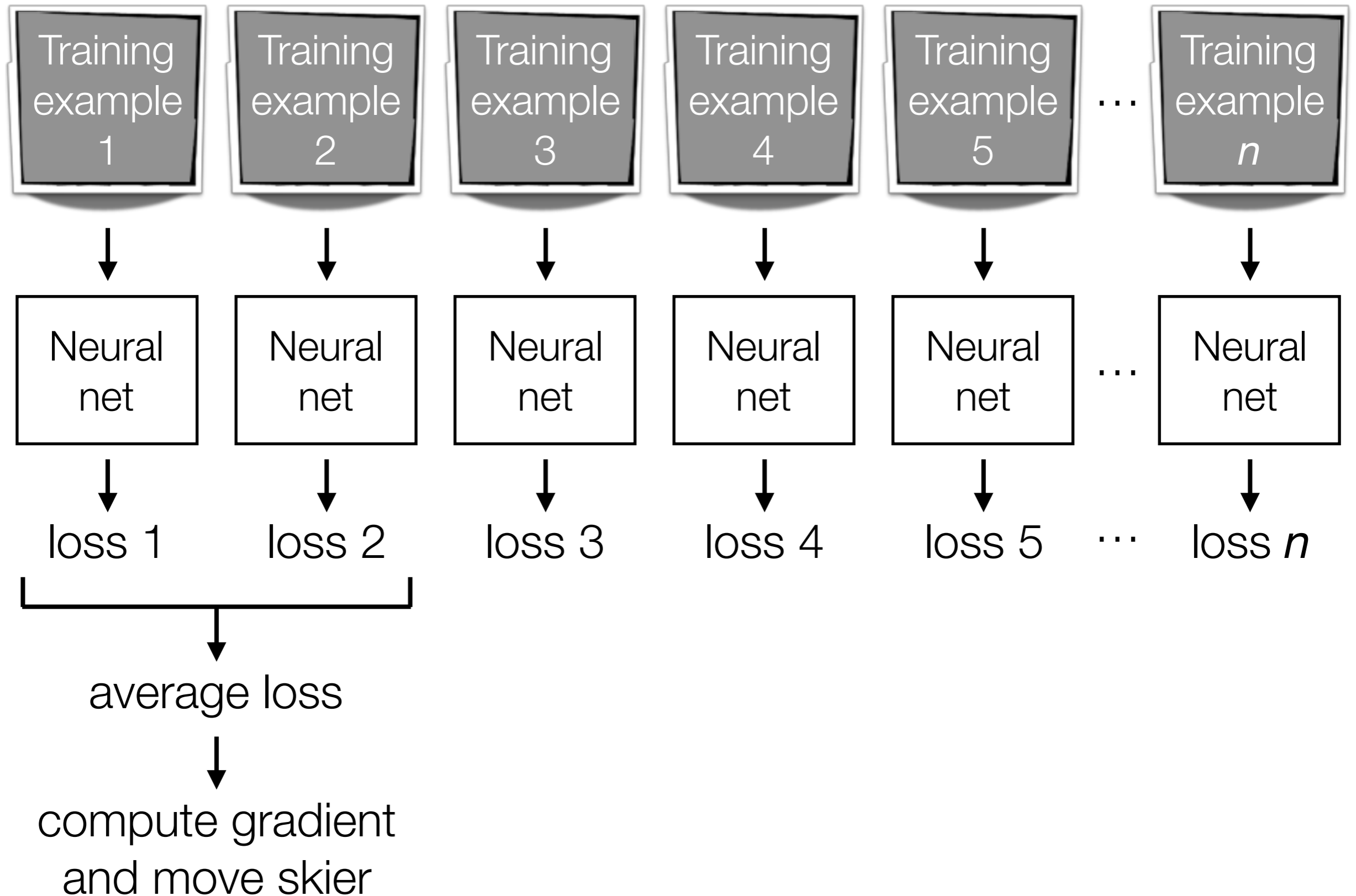


compute gradient
and move skier

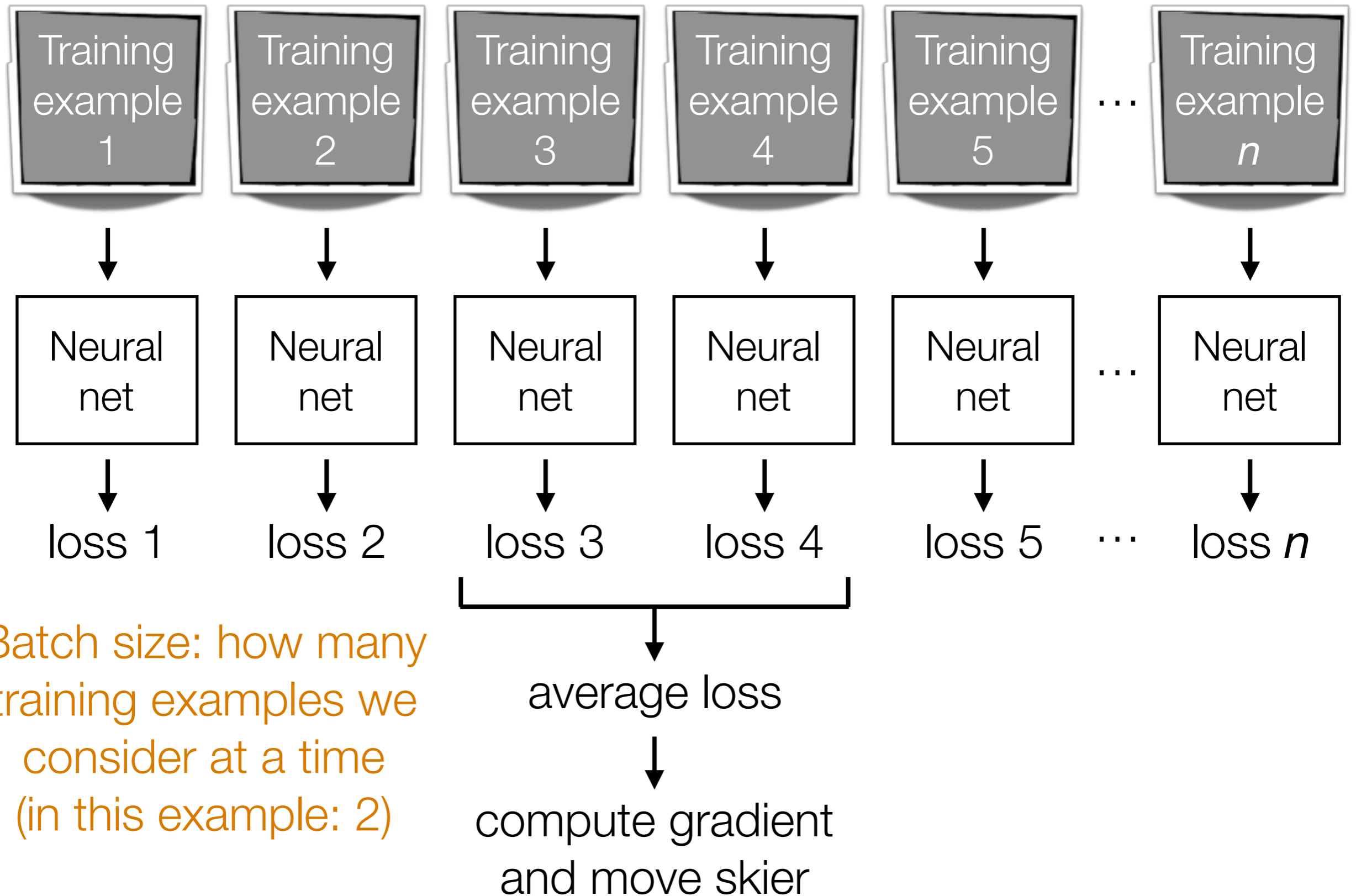
An epoch refers to 1 full pass
through all the training data

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



Batch size: how many training examples we consider at a time (in this example: 2)

Best variant of SGD to use?
Best # of epochs? Best batch size?

Active area of research

Depends on problem, data, hardware, etc

Example: even with a GPU, you can get slow learning (slower than CPU!) if you choose # epochs/batch size poorly!!!

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

Keep in mind: we only covered the super basics!

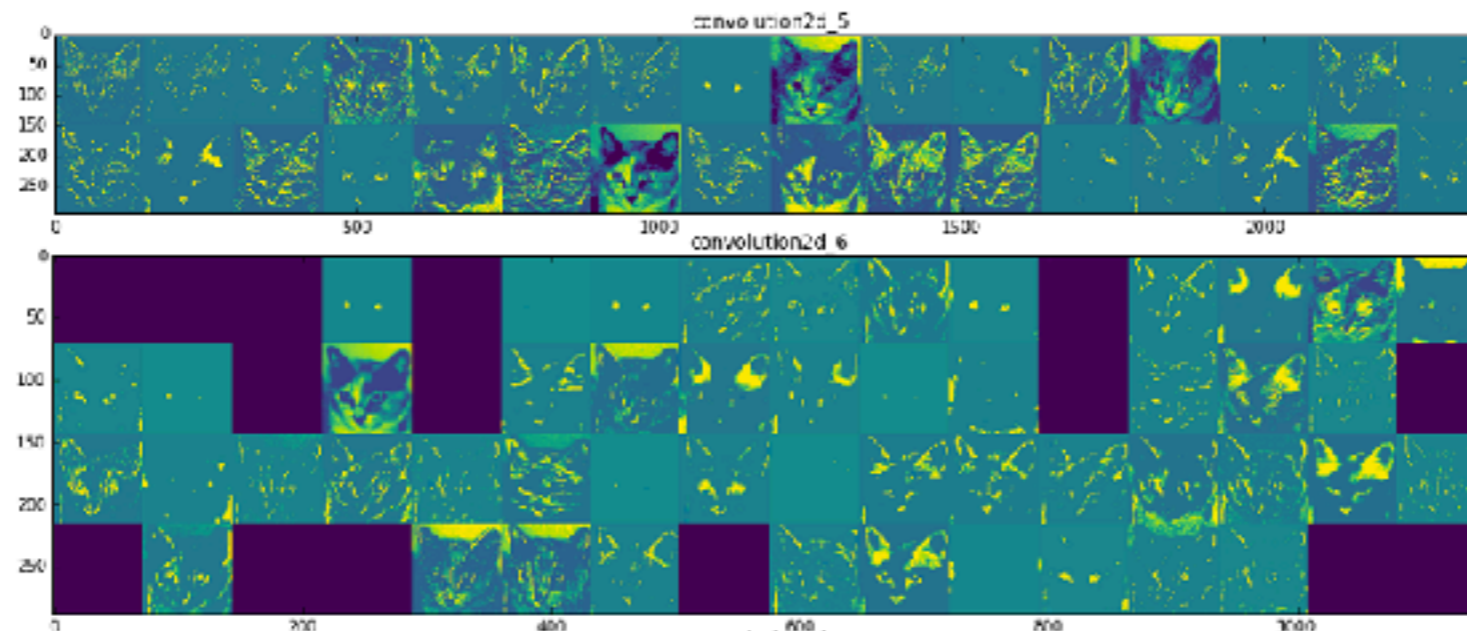
Deep learning has lots of fads!

For example: pooling is on the way out, ResNets are all the rage

Visualizing What a Deep Net Learned

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

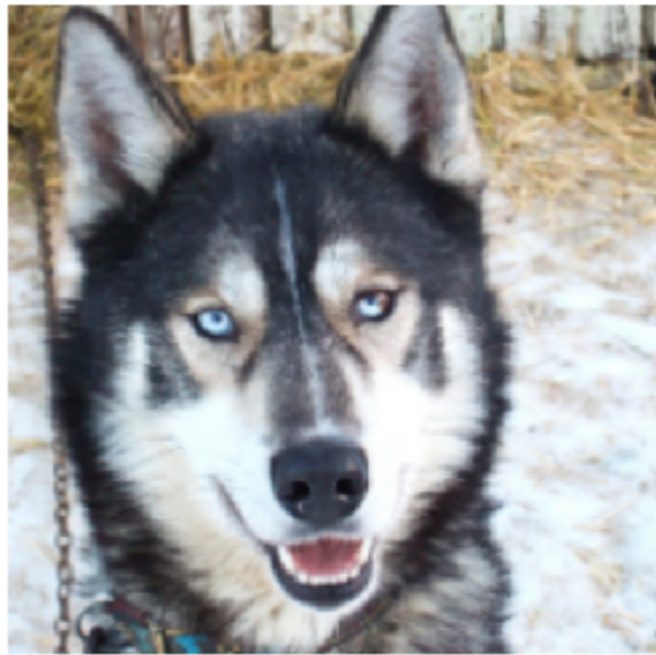
Check course
webpage for
demo



- Plot regions that maximally activate an output neuron



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.

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



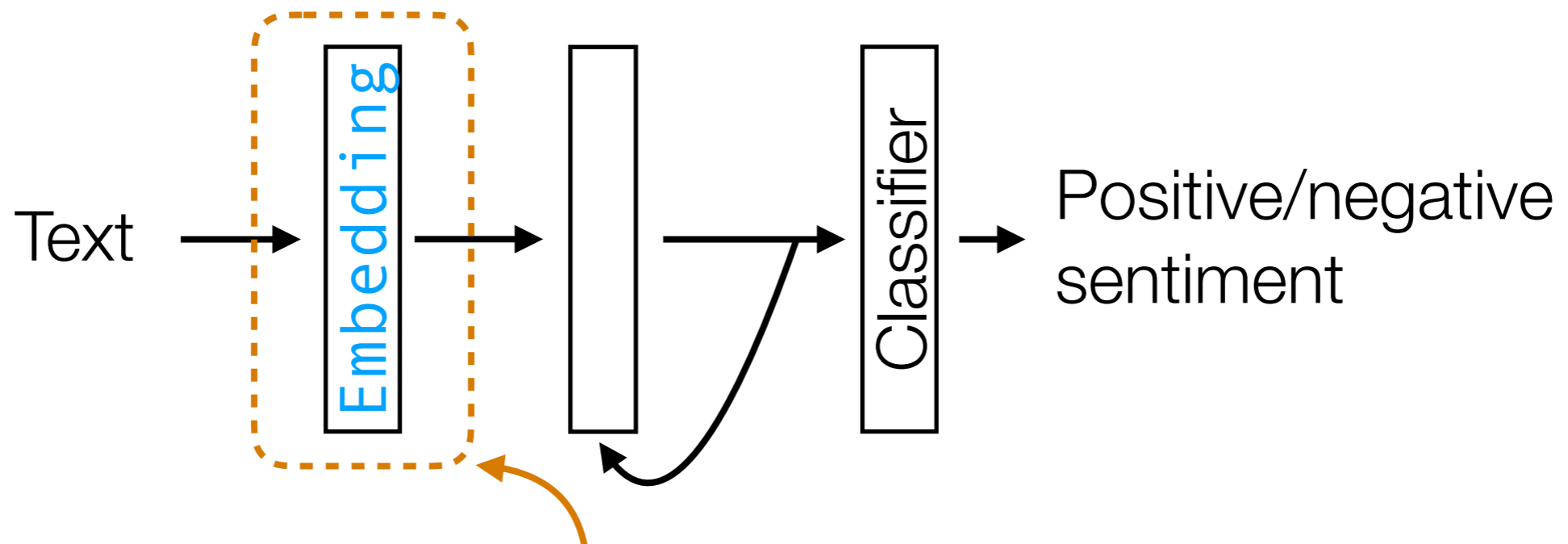
You collect photos from the internet of both, but your dataset size is small, on the order of 1000 images

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!

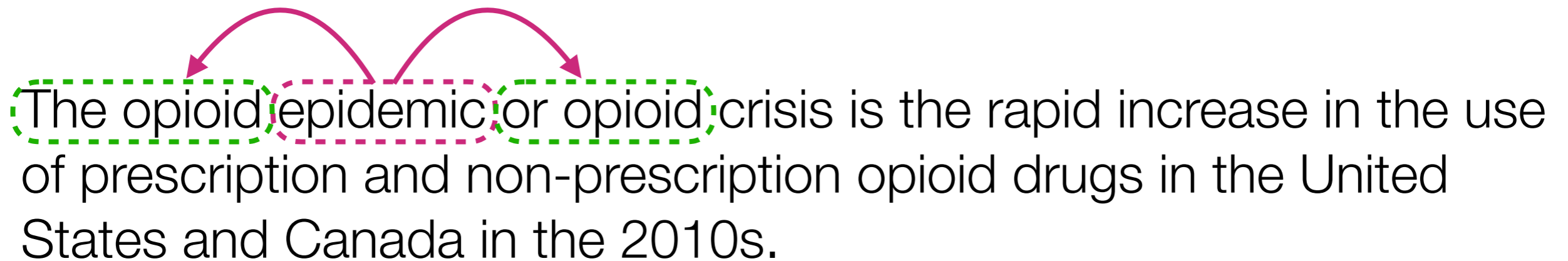
GloVe vectors pre-trained on massive dataset (Wikipedia + Gigaword)

IMDb review dataset is small in comparison

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”

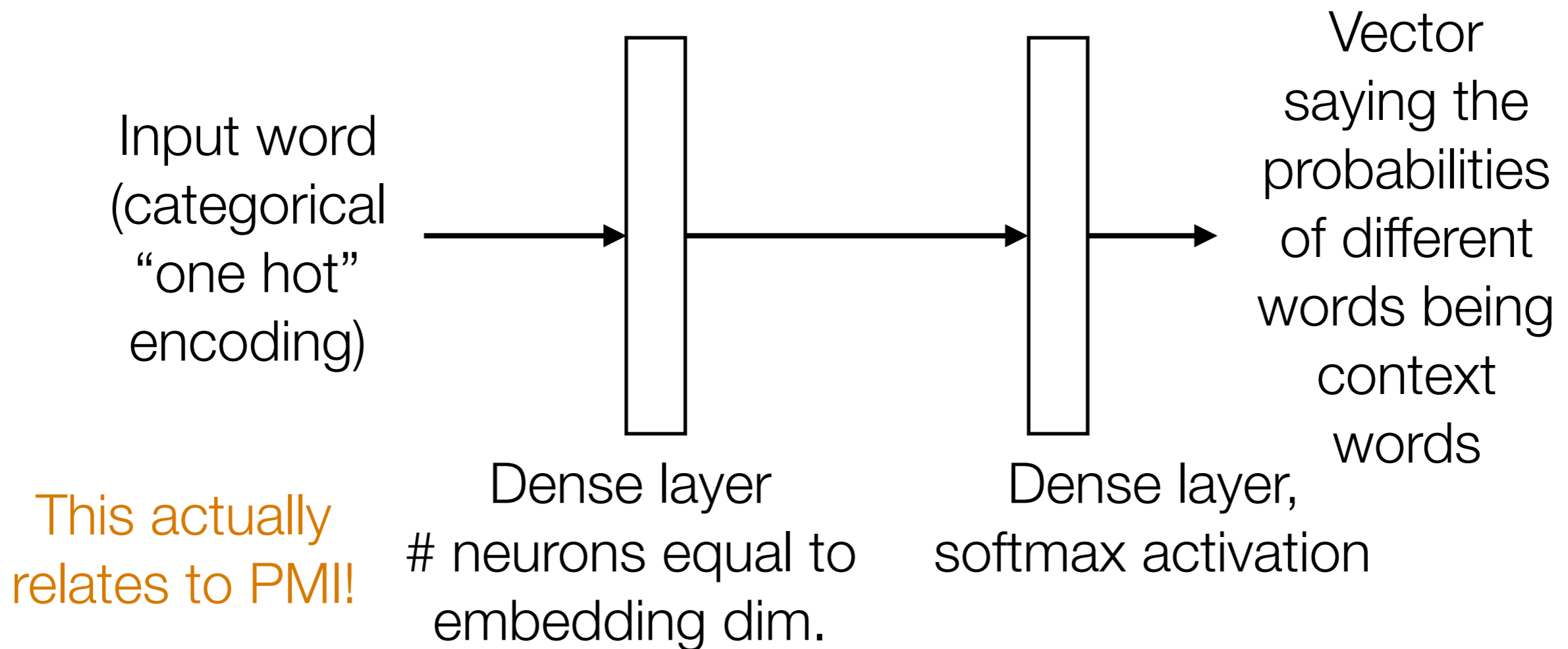


Also provide “negative” examples of words that are *not* likely to be context words (e.g., randomly sample words elsewhere in document)

Self-Supervised Learning

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

Example: word embeddings like word2vec, GloVe



Weight matrix: (# words in vocab) by (embedding dim)

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

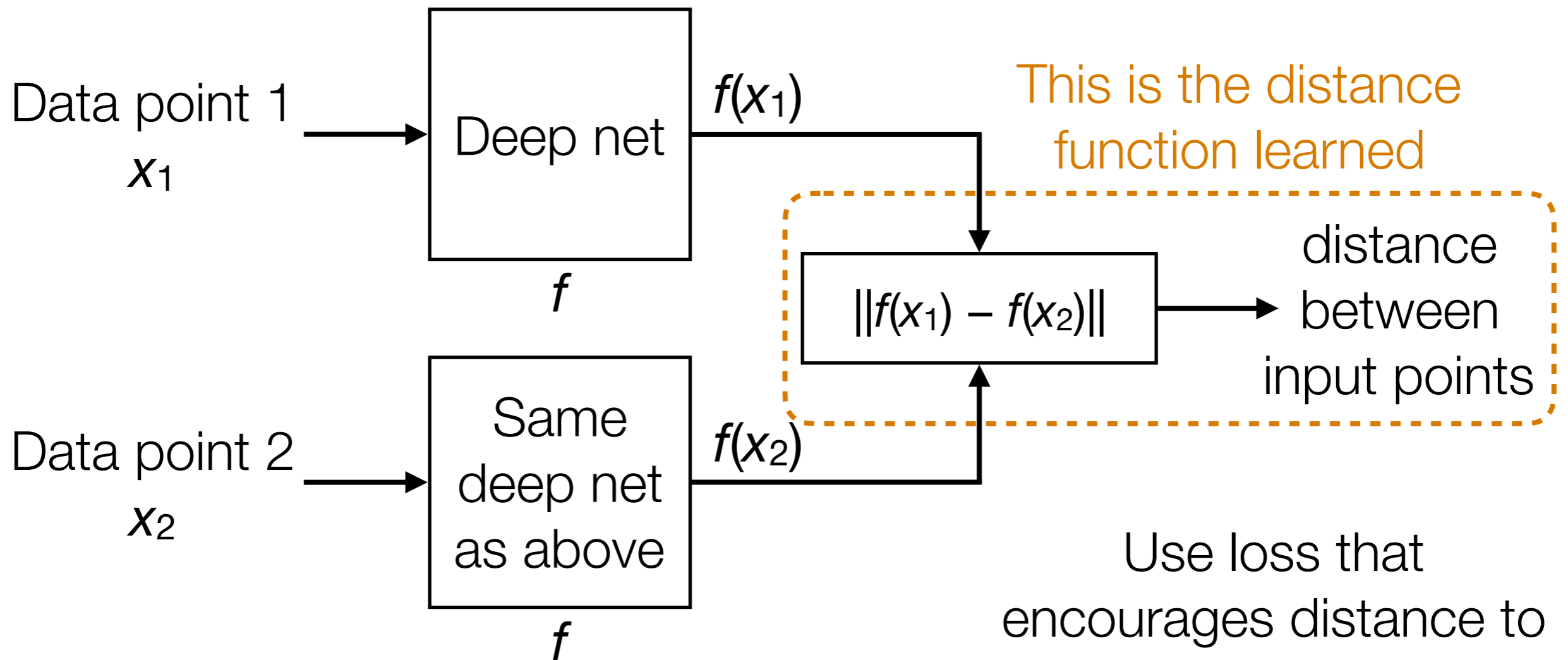
Self-Supervised Learning

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



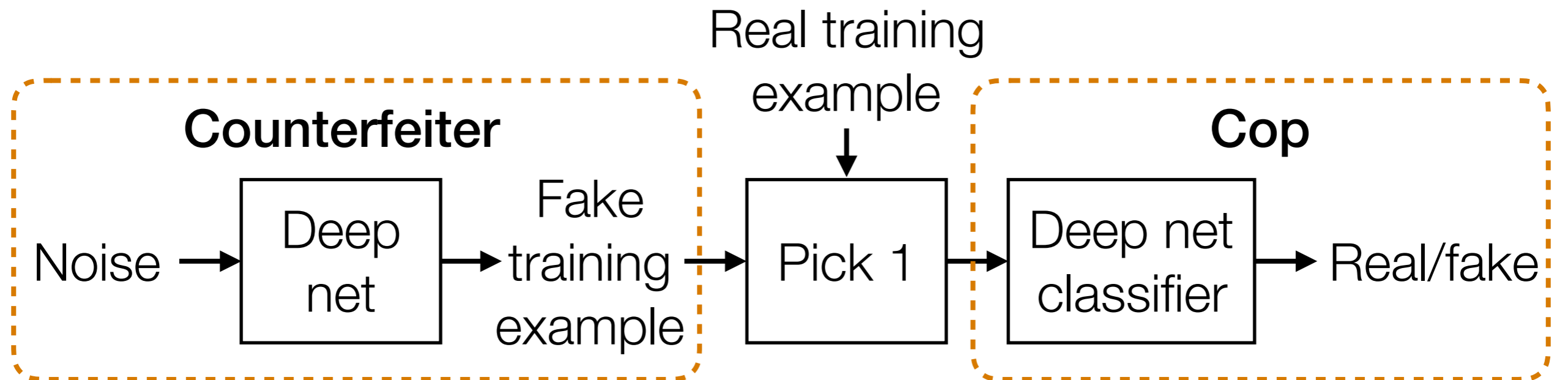
Note: we are learning the function f

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

Monet ↔ Photos



Monet → photo

Zebras ↔ Horses



zebra → horse

Summer ↔ Winter



summer → winter



photo → Monet



horse → zebra



winter → summer



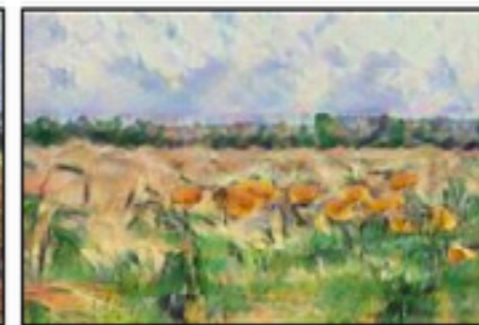
Photograph



Monet



Van Gogh



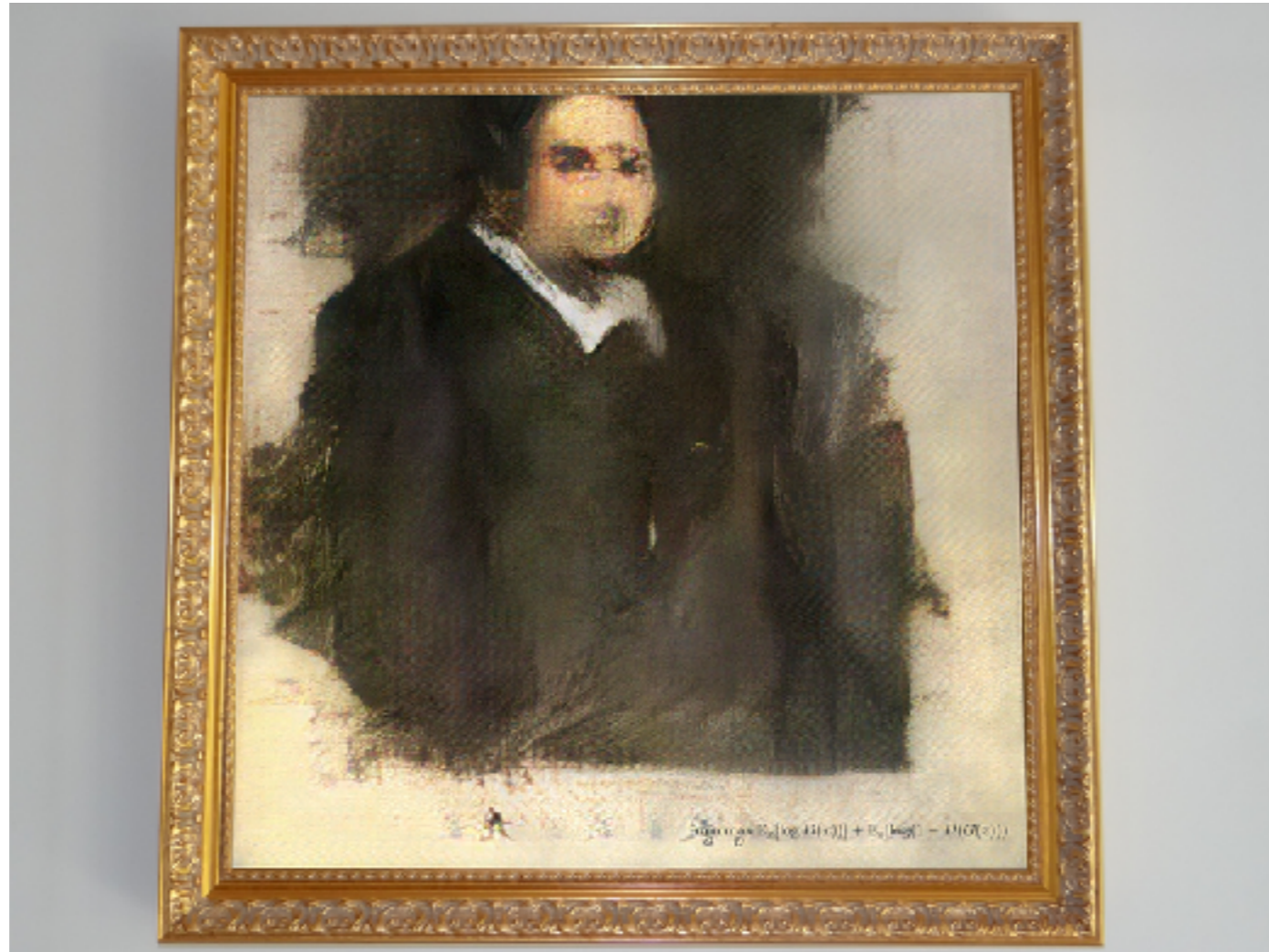
Cezanne



Ukiyo-e

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

Generate Fake Art



October 2018: estimated to go for \$7,000-\$10,000

10/25/2018: Sold for \$432,500

Source: <https://www.npr.org/2018/10/22/659680894/a-i-produced-portrait-will-go-up-for-auction-at-christie-s>

AI News Anchor

China's Xinhua agency unveils AI news presenter

By Chris Baraniuk
Technology reporter

🕒 8 November 2018

f 🗨️ 🐦 ✉️ Share



Source: <https://www.bbc.com/news/technology-46136504>

Harrison Ford as Young Han Solo

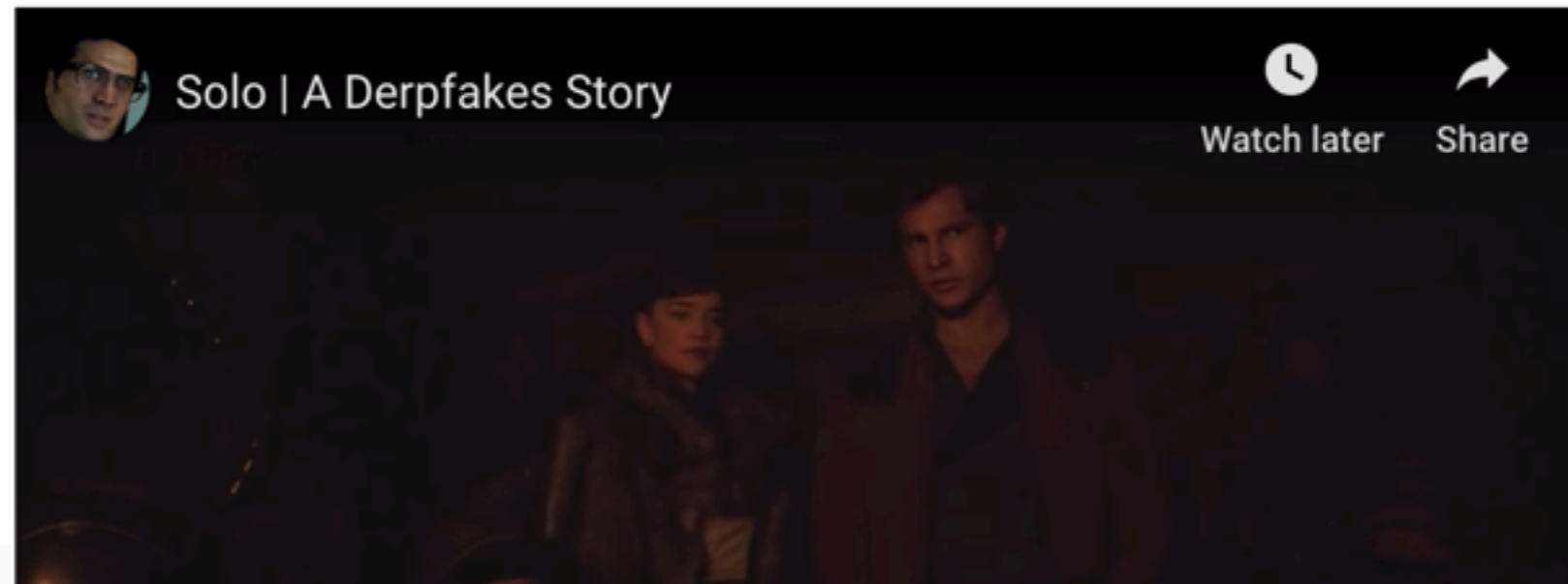
Deepfake edits have put Harrison Ford into Solo: A Star Wars Story, for better or for worse

10 

Uncanny valley, here we come

By [Chaim Gartenberg](#) | [@cgartenberg](#) | Oct 17, 2018, 3:37pm EDT

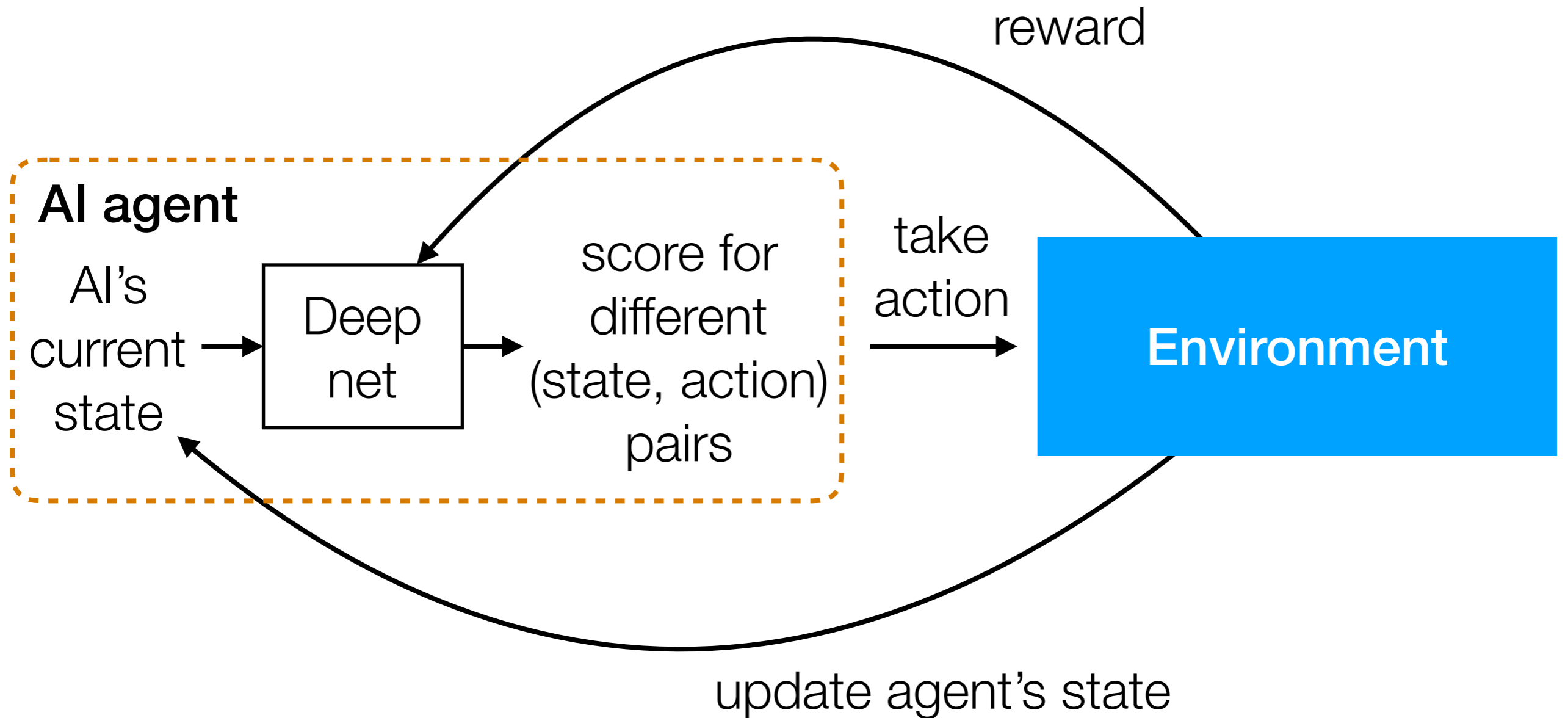
   SHARE



Source: <https://www.theverge.com/2018/10/17/17990162/deepfake-edits-harrison-ford-han-solo-a-star-wars-story-alDEN-ehrenreich>

Deep Reinforcement Learning

The machinery behind AlphaGo and similar systems



Overfitting is Not a Problem?

- In many real-life examples of very large deep nets (lots of parameters): *without regularization*, even when training error goes to 0, validation error keeps decreasing/does not significantly increase!
- Accepted wisdom currently: if you're using an "over-parametrized" deep net for classification, you want to overfit (and get 0 training error)!
 - "Understanding deep learning requires rethinking generalization" (Zhang et al ICLR 2017)
 - Mikhail Belkin at Ohio State University has a bunch of 2018 theory papers that analyze this behavior in which overfitting does not hurt you and instead has good prediction performance

The Future of Deep Learning

- Deep learning currently is still very limited in what it can do — the layers do simple operations and have to be differentiable
 - Adversarial examples at test time remain a problem
 - Doing an elaborate function approximation (curve fitting)
 - The resulting learned function (computer program) is comprised of a series of basic operations, possibly with a `for` loop (for RNN's)
- 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?

[Get started](#)

The deepest problem with deep learning

Some reflections on an accidental Twitterstorm, the future of AI and deep learning, and what happens when you confuse a schoolbus with a snow plow.



Gary Marcus

Dec 1 · 17 min read

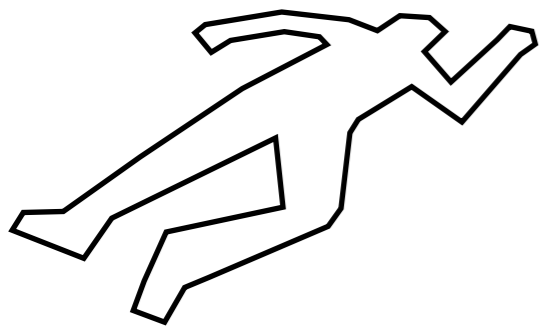
On November 21, I read [an interview with Yoshua Bengio](#) in *Technology Review* that to a surprising degree downplayed recent successes in deep learning, emphasizing instead some other important problems in AI might require important extensions to what deep learning is currently able to do. In particular, Bengio told *Technology Review* that,

I think we need to consider the hard challenges of AI and not be satisfied with short-term, incremental advances. I'm not saying I want to forget deep learning.

Source: <https://medium.com/@GaryMarcus/the-deepest-problem-with-deep-learning-91c5991f5695>

Unstructured Data Analysis

Question



The dead body

This is provided
by a practitioner

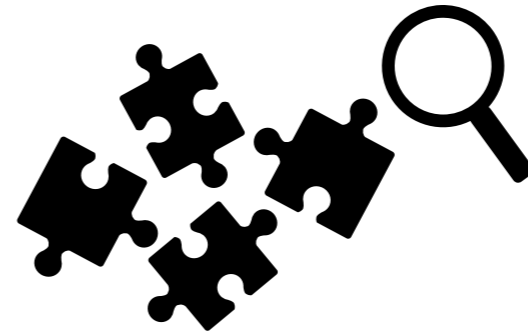
Data



The evidence

Some times you
have to collect
more evidence!

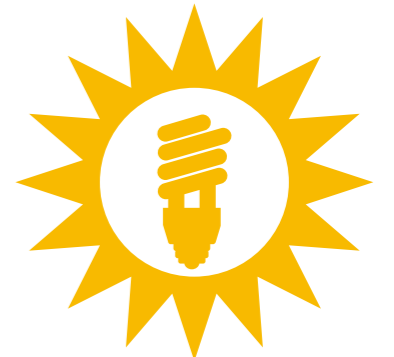
Finding Structure



*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

95-865 Some Parting Thoughts

- Remember to **visualize steps of your data analysis pipeline**
 - Helpful in debugging & interpreting intermediate/final outputs
- 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 (think about how we chose hyperparameters!)**
 - But don't blindly rely on metrics without **interpreting results in the context of your original problem!**
- Often times you won't have labels! If you really want labels:
 - Manually obtain labels (either you do it or crowdsource)
 - Set up self-supervised learning task
- There is a *lot* we did not cover — **keep learning!**