Numerical methods for weather prediction

Machine learning for weather prediction

Christian Lessig

European Centre for Medium-Range Weather Forecasts

What is machine learning?

- Derive rules or structure from data

What is machine learning?

- Derive rules or structure from data
  - Given an image, which object is in the image.

What is machine learning?

- Derive rules or structure from data
  - Given an image, which object is in the image.
  - Given a text, correct the spelling/grammar in it.

What is machine learning?

- Derive rules or structure from data
  - Given an image, which object is in the image.
  - Given a text, correct the spelling/grammar in it.
  - Given an English text, translate it to French.

What is machine learning?

- Derive rules or structure from data
  - Given an image, which object is in the image.
  - Given a text, correct the spelling/grammar in it.
  - Given an English text, translate it to French.
  - Given a description, generate a fitting image.

What is machine learning?

- Derive rules or structure from data
  - Given an image, which object is in the image.
  - Given a text, correct the spelling/grammar in it.
  - Given an English text, translate it to French.
  - Given a description, generate a fitting image.
  - …

What is machine learning?

- Derive rules or structure from data
  - Given an image, which object is in the image.
  - Given a text, correct the spelling/grammar in it.
  - Given an English text, translate it to French.
  - Given a description, generate a fitting image.
  - ...
  - Given an initial condition, generate a weather forecast.

What is machine learning?

- Classical examples of machine learning technique:
  - Interpolation rules (e.g. spline)
  - Linear regression
  - PCA / Karhune-Loeve transform / proper orthogonal decomposition
  - ...

What is machine learning?

- Many different techniques and approaches
  - support vector machines, decision trees, …
- The most prominent approach today are neural networks
  - Historically inspired by psychology and neuroscience as simulations of (human brains)
  - We will look at them from the (natural) vantage point of numerical methods
  - Groundbreaking progress in the last 15-20 years

What is a neural network?

- Neural network is a numerical *nonlinear* mapping:

$$\mathcal{N} : \mathbb{R}^n \to \mathbb{R}^m$$

-

What is a neural network?

- Neural network is a numerical *nonlinear* mapping:

$$\mathcal{N} : \mathbb{R}^n \to \mathbb{R}^m$$

◦

consisting of layers as

$$\mathcal{N} = \mathcal{L}_K \circ \mathcal{L}_{K-1} \circ \cdots \circ \mathcal{L}_1$$

What is a neural network?

- Neural network is a numerical *nonlinear* mapping:

$$\mathcal{N} : \mathbb{R}^n \to \mathbb{R}^m$$

◦

consisting of layers as

$$\mathcal{N} = \mathcal{L}_K \circ \mathcal{L}_{K-1} \circ \cdots \circ \mathcal{L}_1$$

each itself being a mapping

$$\mathcal{L}_k : \mathbb{R}^{n_k} \to \mathbb{R}^{m_k}$$

What is a neural network?

- Linear layer:

$$\mathcal{L}_k = W \in \mathbb{R}^{n_k \times m_k}$$

-

What is a neural network?

- Linear layer:

$$\mathcal{L}_k = W \in \mathbb{R}^{n_k \times m_k}$$

- Multi-layer perceptron

$$\mathcal{L}_k = W_l \cdot \sigma \cdot W_{l-1} \cdots \sigma \cdot W_1$$

where σ is an element-wise nonlinearity, e.g. RELU, sigmoid

What is a neural network?

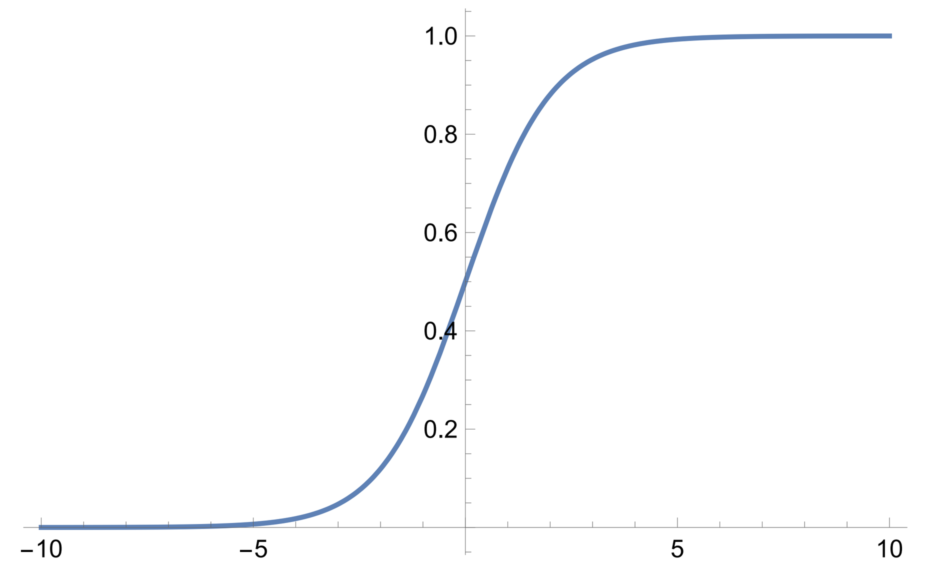
- Linear layer:

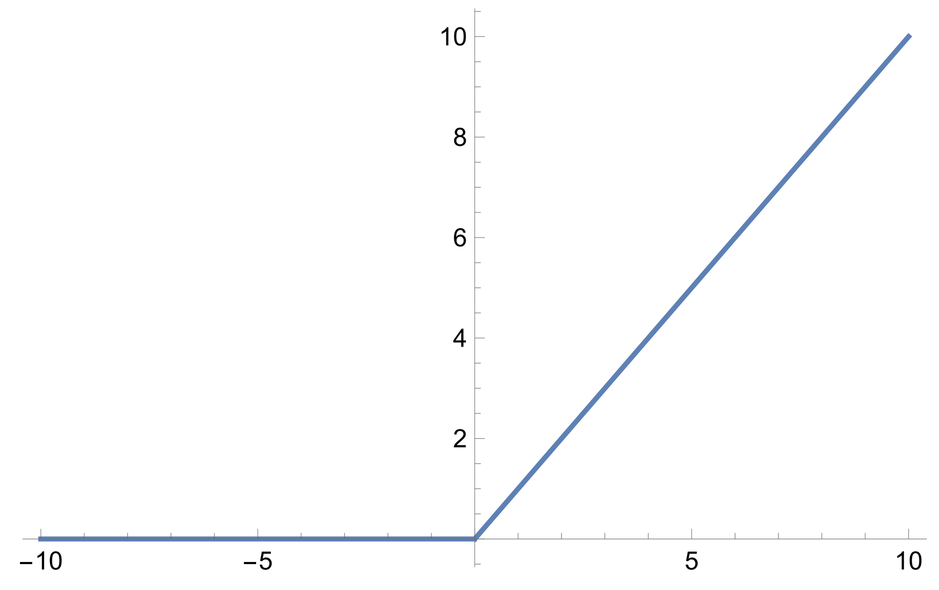
$$\mathcal{L}_k = W \in \mathbb{R}^{n_k \times m_k}$$

- Multi-layer perceptron

$$\mathcal{L}_k = W_l \cdot \sigma \cdot W_{l-1} \cdots \sigma \cdot W_1$$

where σ is an element-wise nonlinearity, e.g. RELU, sigmoid





What is a neural network?

- Linear layer:

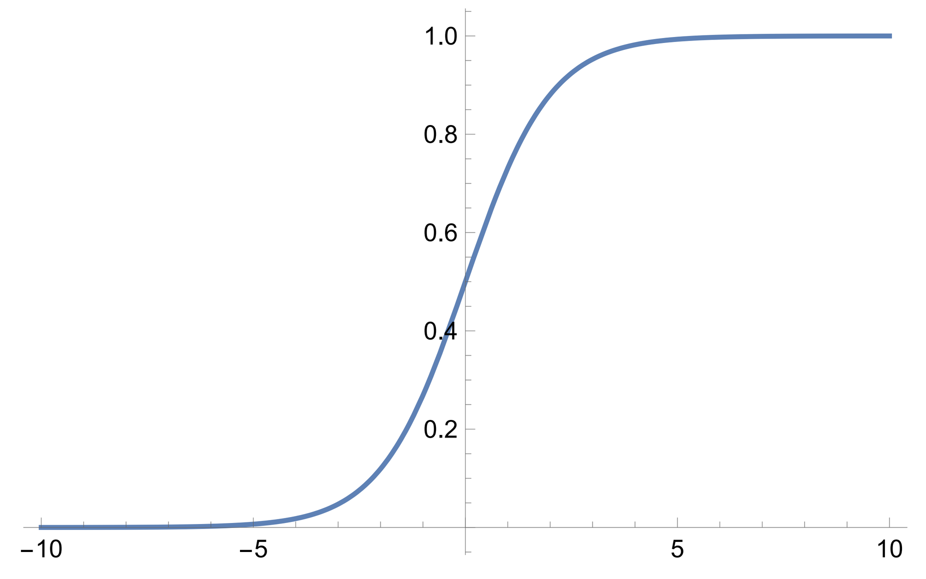
$$\mathcal{L}_k = \boxed{W} \in \mathbb{R}^{n_k \times m_k}$$

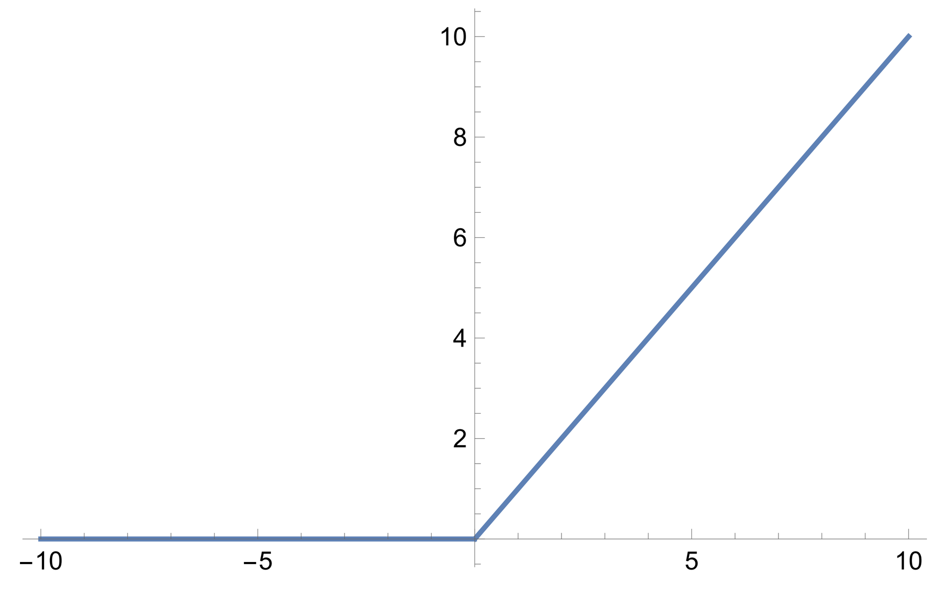
entries are learned, i.e. fitted to the data

- Multi-layer perceptron

$$\mathcal{L}_k = \boxed{W_l} \cdot \sigma \cdot \boxed{W_{l-1}} \cdots \sigma \cdot \boxed{W_1}$$

where σ is an element-wise nonlinearity, e.g. RELU, sigmoid





What is a neural network?

- Weakly nonlinear maps $\mathbb{R}^n \to \mathbb{R}^m$
  - Consisting of simple building blocks
  - Building blocks are largely weight matrices $W \in \mathbb{R}^{n_k \times m_k}$
  - Entries of weight matrices are learned / fitted to data
  - Entirety of learnable weights is denoted as θ

What is a neural network?

- Training:
  - Solution to nonlinear optimization to fit trainable parameters to training data given a loss function ℓ
  - Let $\{(x_i, y_i)\}_{i=1}^{R}$ be a set of training examples
    - x = network input; y = desired network output
  - Training solves in general:

$$\min_\theta L = \min_\theta \frac{1}{R} \sum_{i=1}^{R} \ell(y, \tilde{y}) \ , \ B \ll R$$

What is a neural network?

- Common loss functions:
  - For regression, mean squared error (MSE):

$$\ell(y, \tilde{y}) = \|y - \hat{y}\|_2^2$$

  - For classification, cross entropy loss:

$$\ell(y, \tilde{y}) = \sum_{c=1}^{C} \delta_{y,\tilde{y}} \log(p_{\tilde{y}})$$

known label is interpreted a discrete Kronecker prob. distribution $\delta_{y,\tilde{y}}$

What is a neural network?

- Nonlinear optimization problem is typically solved with stochastic gradient descent (or a variant of it such as ADAM)

$$\nabla L \approx \frac{1}{B}\sum_{i=1}^{B} \nabla \ell(y, \tilde{y}) \ , \ B \ll R$$

What is a neural network?

- Nonlinear optimization problem is typically solved with stochastic gradient descent (or a variant of it such as ADAM)

$$\nabla L \approx \frac{1}{B} \sum_{i=1}^{B} \nabla \ell(y, \tilde{y}) \ , \ B \ll R$$

  - Gradient is approximated using a small number of examples, a so called batch
  - Makes optimization computationally tractable but also improves robustness to local minima

What is a neural network?

```
class NeuralNetwork(nn.Module):
    def __init__(self):
        super().__init__()
        self.flatten = nn.Flatten()
        self.linear_relu_stack = nn.Sequential(
            nn.Linear(28*28, 512),
            nn.ReLU(),
            nn.Linear(512, 512),
            nn.ReLU(),
            nn.Linear(512, 10),
        )

    def forward(self, x):
        x = self.flatten(x)
        logits = self.linear_relu_stack(x)
        return logits
```

https://pytorch.org/tutorials/beginner/pytorch_with_examples.html

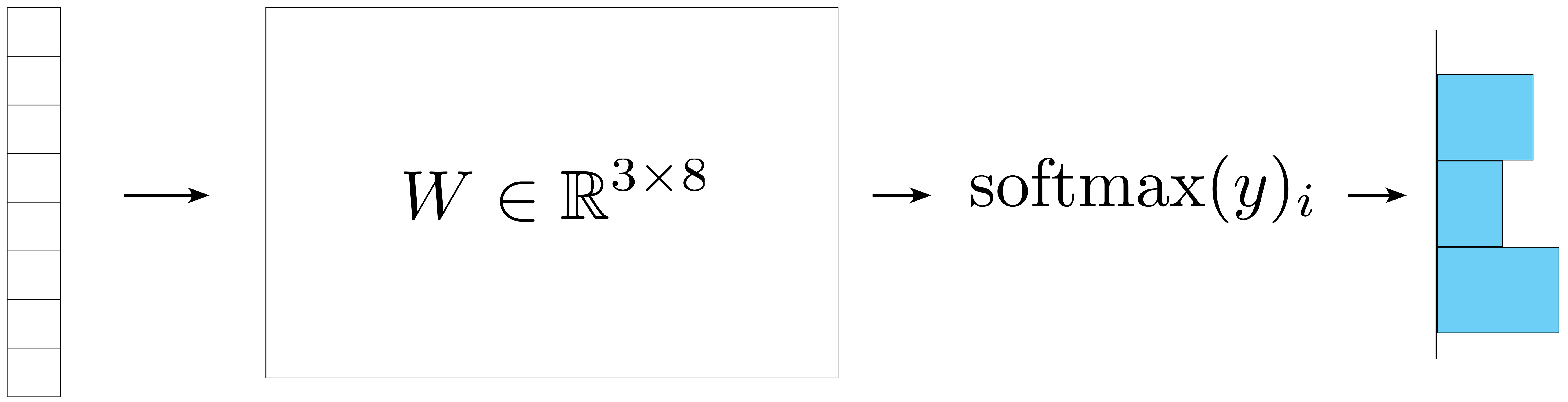
What is a neural network?

```python
# The nn package also contains definitions of popular loss functions; in this
# case we will use Mean Squared Error (MSE) as our loss function.
loss_fn = torch.nn.MSELoss(reduction='sum')

learning_rate = 1e-6
for t in range(2000):

    # Forward pass: compute predicted y by passing x to the model. Module objects
    # override the __call__ operator so you can call them like functions. When
    # doing so you pass a Tensor of input data to the Module and it produces
    # a Tensor of output data.
    y_pred = model(xx)

    # Compute and print loss. We pass Tensors containing the predicted and true
    # values of y, and the loss function returns a Tensor containing the
    # loss.
    loss = loss_fn(y_pred, y)
    if t % 100 == 99:
        print(t, loss.item())

    # Zero the gradients before running the backward pass.
    model.zero_grad()

    # Backward pass: compute gradient of the loss with respect to all the learnable
    # parameters of the model. Internally, the parameters of each Module are stored
    # in Tensors with requires_grad=True, so this call will compute gradients for
    # all learnable parameters in the model.
    loss.backward()

    # Update the weights using gradient descent. Each parameter is a Tensor, so
    # we can access its gradients like we did before.
    with torch.no_grad():
        for param in model.parameters():
            param -= learning_rate * param.grad
```

https://pytorch.org/tutorials/beginner/pytorch_with_examples.html

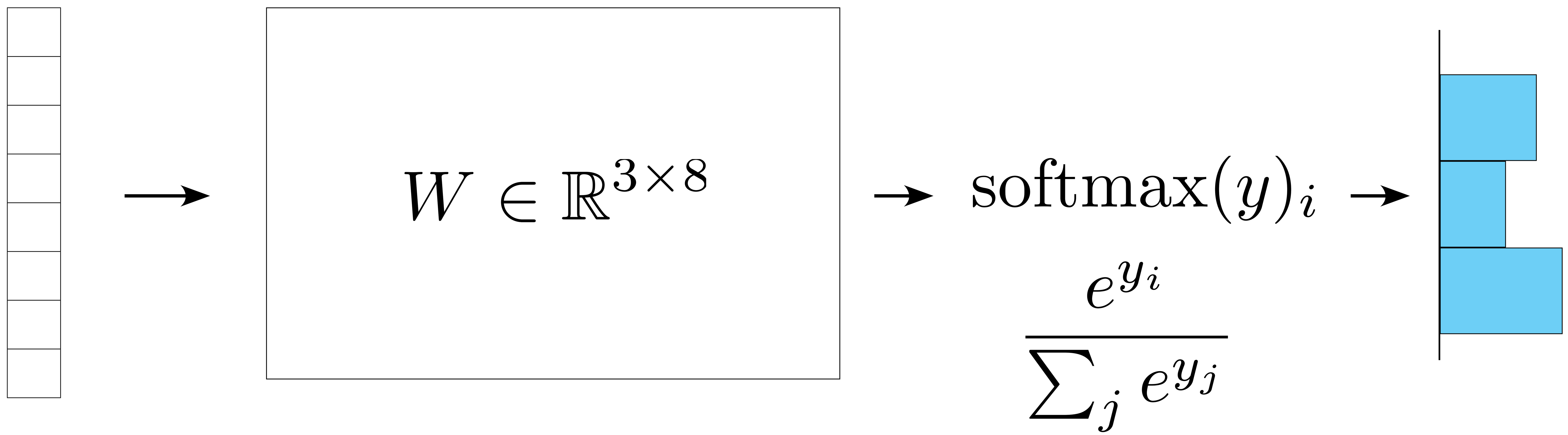
Advanced neural network architectures

- Prediction heads for classification problems
  - Most intuitive approach for prediction a class label c_i is to directly predict the label (perhaps thresholded)
  - In practice, a probability over all possible classes is predicted



Advanced neural network architectures

- Prediction heads for classification problems
  - Most intuitive approach for prediction a class label c_i is to directly predict the label (perhaps thresholded)
  - In practice, a probability over all possible classes is predicted



Generative machine learning

- Generate data from a (potentially conditional) probability distribution

$$p_\theta(y|x) \approx p(y|x)$$

Generative machine learning

- Generate data from a (potentially conditional) probability distribution

$$\underbrace{p_\theta(y|x)}_{\mathcal{N}_\theta(x)} \approx p(y|x)$$

Generative machine learning

- Generate data from a (potentially conditional) probability distribution

$$\underbrace{p_\theta(y|x)}_{\mathcal{N}_\theta(x)} \approx p(y|x)$$

  - Discrete: probabilistic prediction
  - Regression (continuous): e.g. diffusion models

Advanced neural network architectures

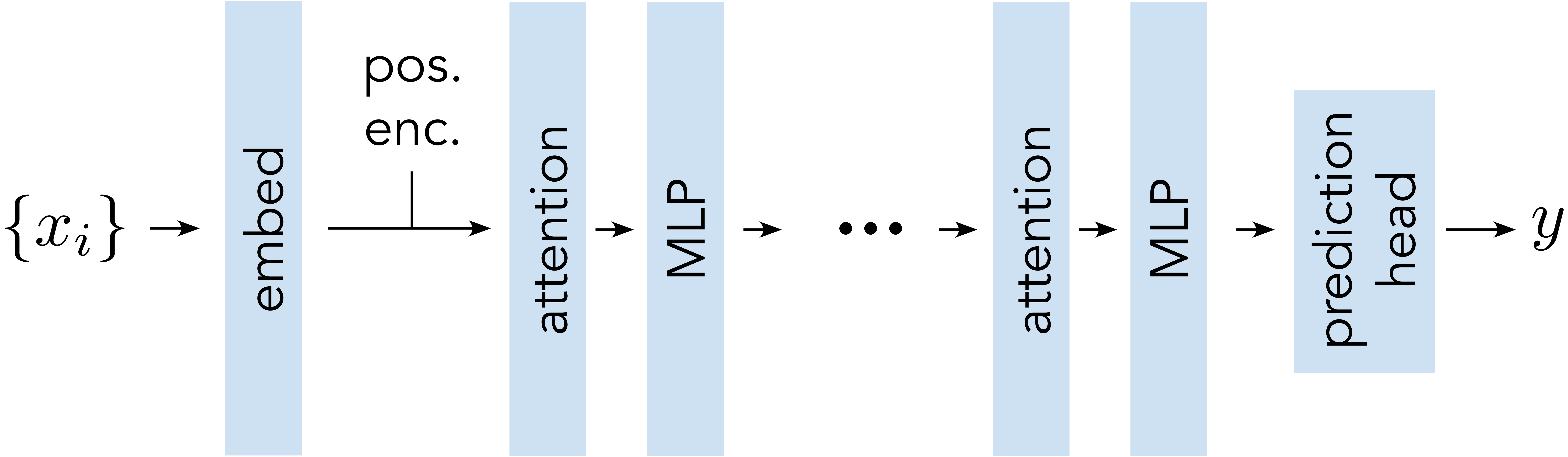
- Transformer neural networks
  - Input is set/sequence of vectors x_i (e.g. from words)
  - Self-attention computes similarity between latent representation of vectors and updates based these on this

$$q_i = W_q x_i \quad k_i = W_k x_i \quad v_i = W_v x_i$$

$$\bar{q}_i = \sum_j \sigma(q_i \cdot k_j) v_j$$

Advanced neural network architectures

- Transformer neural networks



Advanced neural network architectures

- Convolutional neural networks
  - Central building block are learnable convolutions
- Graph neural networks
  - Similar to transformer but with graph to structure information exchange between latent space representations
- U-Net
  - Network has U-like shape with decreasing/increasing dimension to have multiple levels of abstraction

Advanced neural network architectures

Our work reinforces the bitter lesson. The most important factors determining the performance of a sensibly designed model are the compute and data available for training[5]

https://arxiv.org/pdf/2310.16764

Advanced neural network architectures

Our work reinforces the bitter lesson. The most important factors determining the performance of a sensibly designed model are the compute and data available for training[5]

https://arxiv.org/pdf/2310.16764

[5] By sensibly designed, we mean models that are sufficiently expressive and have stable gradient propagation.

Summary

- Machine learning: derive rules or structure from data
- Neural networks
  - Weakly nonlinear mappings between real vector spaces with matrix entries as trainable parameters
  - Parameters are “fit“ using stochastic gradient descent
  - Allows to effectively solve nonlinear optimization problems with billions of free parameters