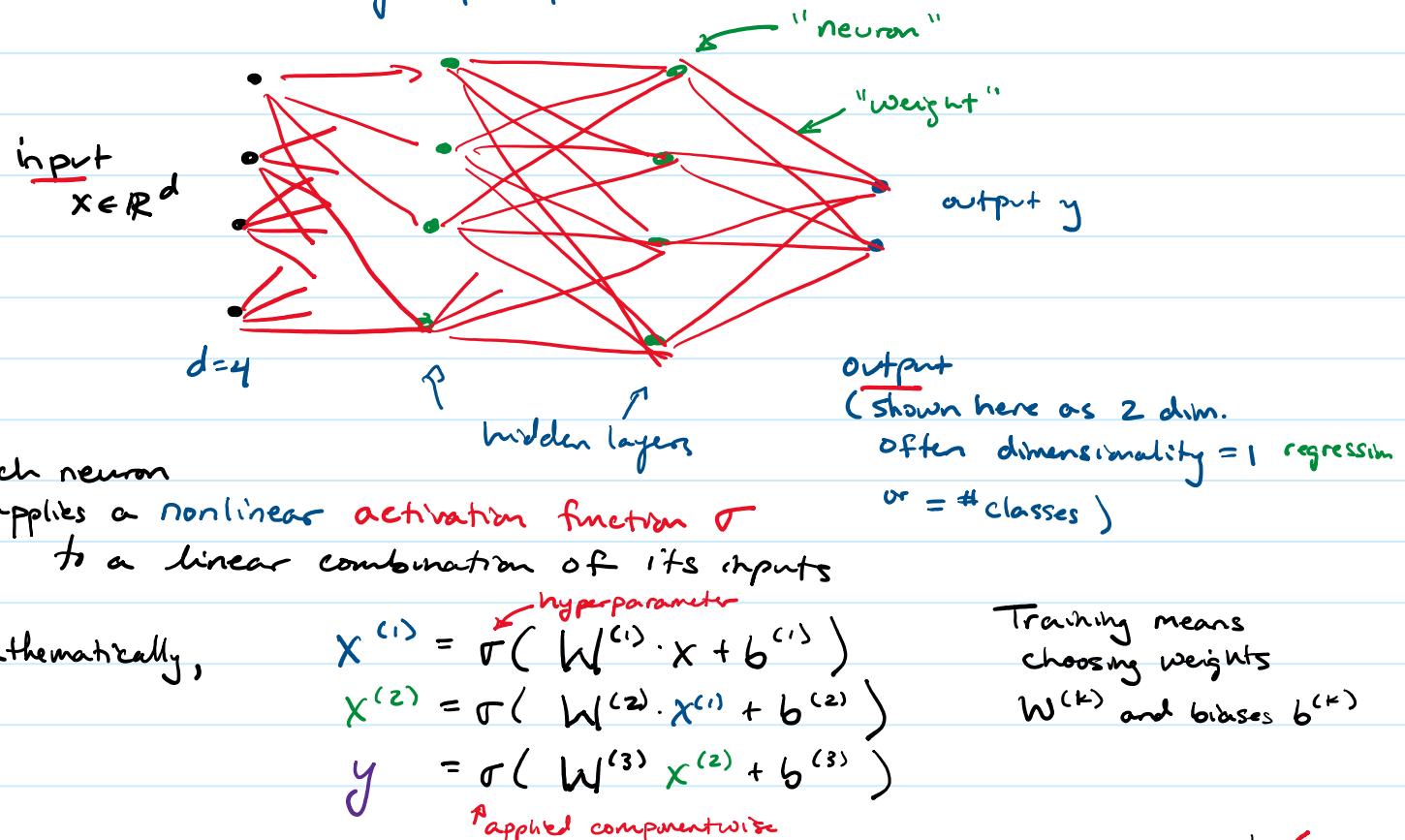


(Artificial) Neural Networks

Sunday, August 25, 2024 9:56 AM

Neural Nets simplest kind is "fully-connected feed-forward" aka "multi-layer perceptron" MLP



Activation functions: ReLU (rectified linear unit), $\sigma(a) = \max(a, 0)$ ✓
Sigmoid, "Gelu", ... also common choices

Approximation Error

https://en.wikipedia.org/wiki/Stone%20-%20Weierstrass_theorem

- Weierstrass

The set of polynomials is dense in $C([a,b])$ w.r.t. $\| \cdot \|_\infty$
i.e. $\forall f \in C([a,b])$, $\forall \varepsilon > 0$, \exists polynomial p s.t. $|p(x) - f(x)| < \varepsilon \quad \forall x \in [a,b]$

Universal Approximation Thm

A spate of papers in the 1980s—1990s, from George Cybenko and Kurt Hornik etc, established several universal approximation theorems for arbitrary width and bounded depth. [37][3][38][4] See [39][40][6] for reviews. The following is the most often quoted:

Universal approximation theorem — Let $C(X, \mathbb{R}^m)$ denote the set of continuous functions from a subset X of a Euclidean \mathbb{R}^n space to a Euclidean space \mathbb{R}^m . Let $\sigma \in C(\mathbb{R}, \mathbb{R})$. Note that $(\sigma \circ x)_i = \sigma(x_i)$, so $\sigma \circ x$ denotes σ applied to each component of x .

Then σ is not polynomial if and only if for every $n \in \mathbb{N}$, $m \in \mathbb{N}$, compact $K \subseteq \mathbb{R}^n$, $f \in C(K, \mathbb{R}^m)$, $\varepsilon > 0$ there exist $k \in \mathbb{N}$, $A \in \mathbb{R}^{k \times n}$, $b \in \mathbb{R}^k$, $C \in \mathbb{R}^{m \times k}$ such that

$$\sup_{x \in K} \|f(x) - g(x)\| < \varepsilon$$

where $g(x) = C \cdot (\sigma \circ (A \cdot x + b))$

Fancy Neural Networks

Sunday, August 25, 2024 4:41 PM

Convolutional Layers

Basic idea: require the weights W to be a convolution (in 1D, 2D, ...)

Why?

- Fewer parameters
- Can prevent overfitting
- Can speed up code
- Often makes sense due to prior knowledge
very common in image applications
- Plain convolution is **translation equivariant**
i.e. translate input \rightarrow translate output **Equivariance**
(translate input \rightarrow no effect) **Invariance**)

Input: $X \in \mathbb{R}^d$

Output: $y \in \mathbb{R}^m$ Fully connected: $y = W \cdot x$

$$\boxed{W}^d$$

$m \cdot d$ parameters

Simplest convolution: w_i aka Kernel filter $h \in \mathbb{R}^d$,

Output is size $m \approx d$
($m = d$ if periodic)

$$y = h * x \quad y_i = \sum_j x_{i-j} \cdot h_j$$

So convolution has d parameters
instead of d^2 (for fully connected)

$$= \sum_j x_j h_{i-j}$$

I'm ignoring
"boundary conditions"

Typically h only has a few nonzeros (kernel size),

Say k , so in fact convolution has then just k parameters
and takes just $O(k)$ flops

Practical Considerations

• Usually have several filters "in parallel", creating new "channels"

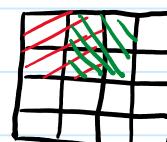
• Parameters per kernel:

• padding / boundary conditions

• Stride

• filter size

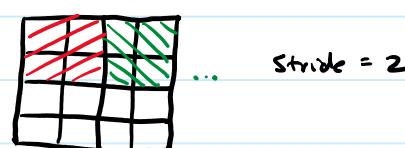
2D conv, size 2×2



... Stride = 1

• Often followed by a
pooling operation

(e.g. max-pooling) to
downsample (must choose
size)

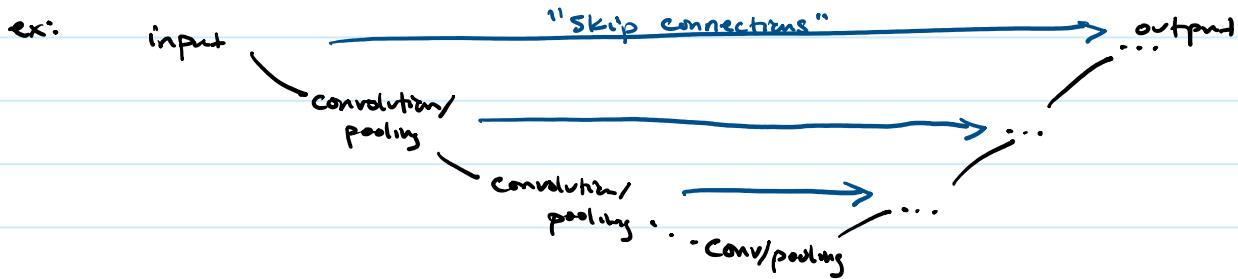


... Stride = 2

Fancy Neural Networks: U-nets, Resnets, AutoEncoders

Tuesday, August 27, 2024 4:26 PM

U-nets A type of CNN often used in medical imaging



Resnets (Residual Networks) / skip connections

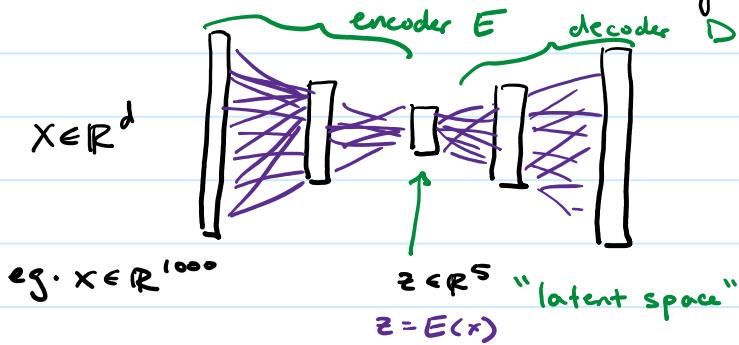
Instead of $x^{(k+1)} = \sigma(W \cdot x^{(k)} + b)$

do $x^{(k+1)} = \underbrace{x^{(k)}}_{\text{identity}} + \underbrace{\sigma(W \cdot x^{(k)} + b)}_{\text{perturbation to identity}}$

Sometimes do $x^{(k-3)}$ or something

Helps "stabilize" training if used appropriately, can help w/ vanishing gradients

Autoencoders (AE) have a **bottleneck** layer



AE are self-supervised:

$$\text{loss is } \|D(E(x)) - x\|^2$$

ideally $D \circ E$ is identity
(but it can't be that
for all inputs due to
bottleneck, i.e. $s < d$)

We'll later talk about

Variational AE (VAE) a type of generative model
(as well as **GANS** and diffusion models)

Fancy Neural Networks: Recurrent and Transformers

Tuesday, August 27, 2024 4:46 PM

Recurrent Neural Nets (ex: LSTM)

Helps with two (related) issues: 1) what if not all X are the same size?

2) how to encode positional structure?

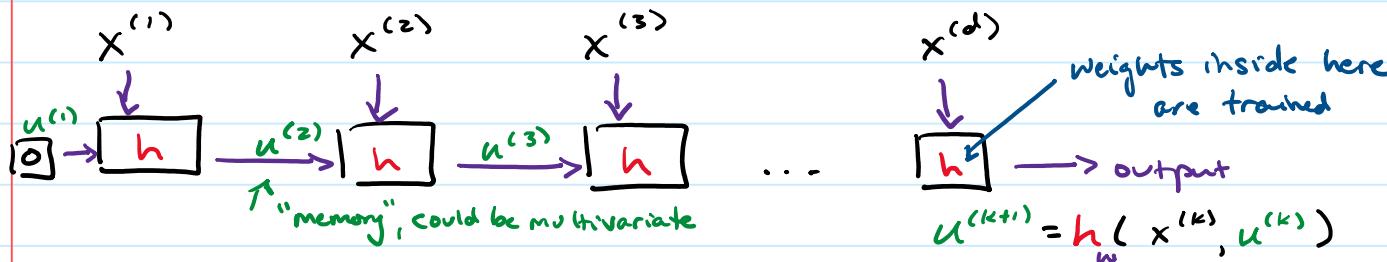
i.e. standard ML: $X_i = \begin{bmatrix} \# \text{ typos in email} \\ \# \text{ all capital words in email} \end{bmatrix} \in \mathbb{R}^2$ ↪ position doesn't matter

vs. time-series ML: $X_i = \text{(the entire email)}$ ↪ position matters

Ex: NLP, time-series (robotics, guidance-navigation-control)

RNN deal with this using special layers.

Let $X \in \mathbb{R}^d$ ↪ "d" depends on X (ex: $X = [\text{H e l l o_w o r l d}]$)



Transformers and Attention mechanism "Attention is all you need" 2017

Also popular in NLP

Vaswani et al. 10⁴ + citations

Encoding

→ old-school: "bag of words", "one hot"

→ large: digrams, trigrams { "He", "el", "ll", "lo", "o-", ... }
large!

→ RNN style

→ positional encodings: word encoded as a vector
(word embedding)

and location in sentence encoded as vector

Index of Token Positional encoding

Index of Token	Positional encoding	d is parameter you choose
Hello	$P_{0,0} \ P_{0,1} \ \dots \ P_{0,d}$	
World	$P_{1,0} \ P_{1,1} \ \dots \ P_{1,d}$	
today.	$P_{2,0} \ P_{2,1} \ \dots \ P_{2,d}$	

$$P_{k,2i} = \sin\left(\frac{k}{n^{2/d}}\right), \quad P_{k,2i+1} = \cos\left(\frac{k}{n^{2/d}}\right) \quad n = 10^4 \text{ or parameter you set}$$

Fancy Neural Networks: Transformers

Wednesday, September 4, 2024 2:14 PM reference: paper/website of Davoud Atare Tarzanagh

after tokenizing a sequence (still a single "datum") of length T
now have a matrix $\begin{matrix} T \\ | \\ X \end{matrix}$ where d = dimensionality of each "token"

transformers apply self-attention then MLP layers
main innovation

Self-attention layer

$$X \mapsto S(XQ \cdot K^T X^T) \cdot Xv$$

↑ don't "flatten" X ,
keep matrix structure

Softmax nonlinearity

X_Q "queries"
 X_K^T "keys"
 X_V "values"

$Q, K, V \in \mathbb{R}^{d \times d}$ are trainable
sometimes $V \in \mathbb{R}^{d \times 1}$

$$S(v)_t = \frac{e^{v_t}}{\sum_{s=1}^T e^{v_s}}$$

motivation: selects a few most relevant tokens

Other fancy neural nets

see d2l.ai "Dive into Deep Learning" Zhang et al.

up-to-date, broad support, H2O, code

or "Deep Learning" ('24, Springer) Christopher Bishop

or deeplearningbook.org (Goodfellow, Bengio, Courville) not as up-to-date

In this class, we'll cover advanced models as needed

ex: VAE, GANs, Diffusion Models, Neural Radiance Field (NeRF),

INR, Hypernetworks, Neural Operator, PINO, PINN, DeepONet, ...