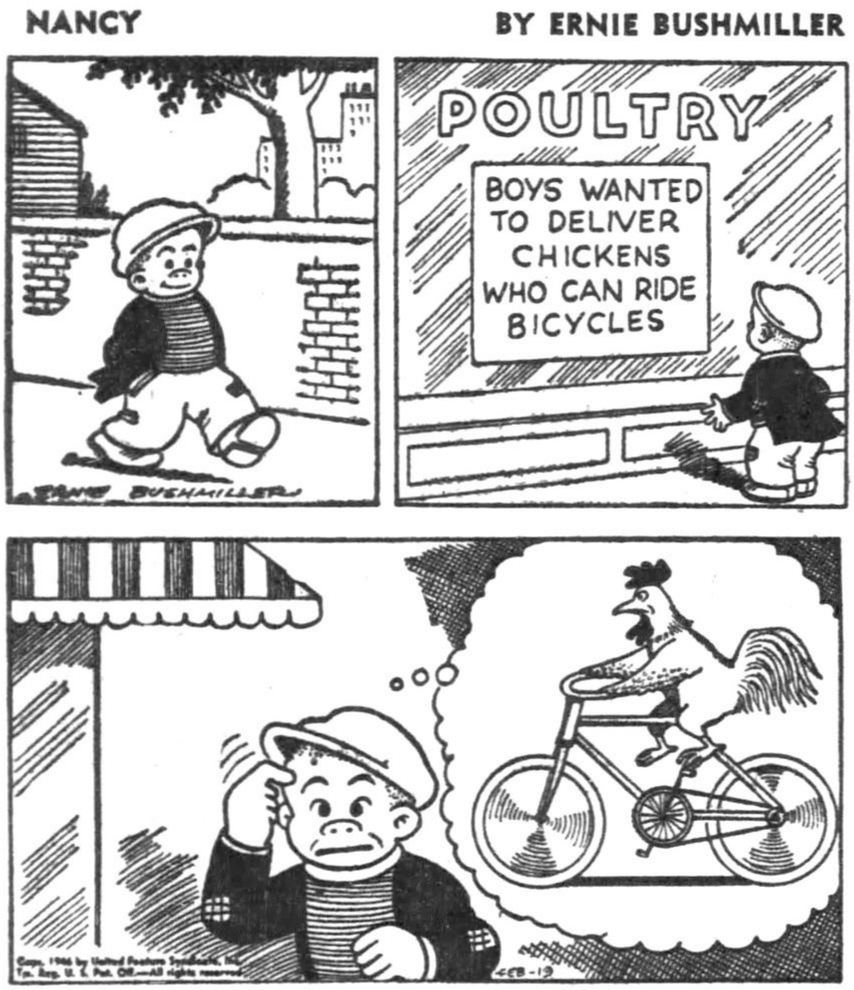
ANNOUNCEMENTS

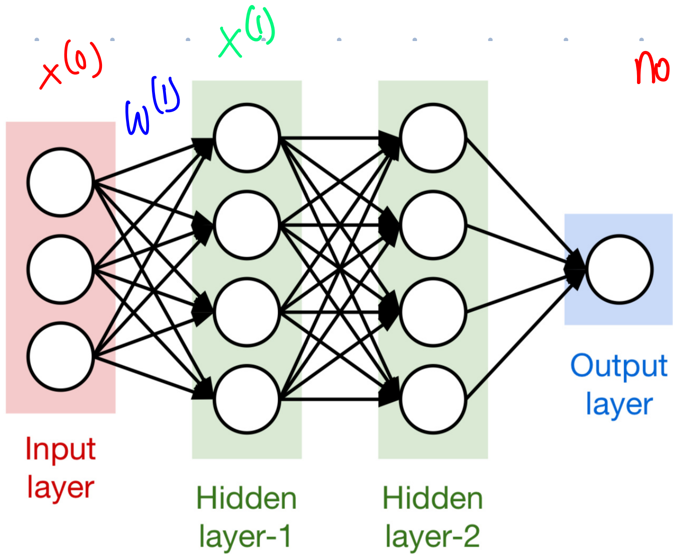
1. All assignments, long and short, optional and non-optional are released
2. Kaggle competition + dataset released; "BASELINES" out soon!



Last few lectures ...

I. FULLY-CONNECTED NETWORK

$$\underbrace{x^{(l)} = g}_{\text{non-linearity}}\left(W^{(l)} x^{(l-1)} + b^{(l)}\right)$$



II. RESIDUAL CONNECTIONS

$$x \longrightarrow f(\cdots) \longrightarrow f(x) \underbrace{+\, x}_{\text{Residual connection}}$$

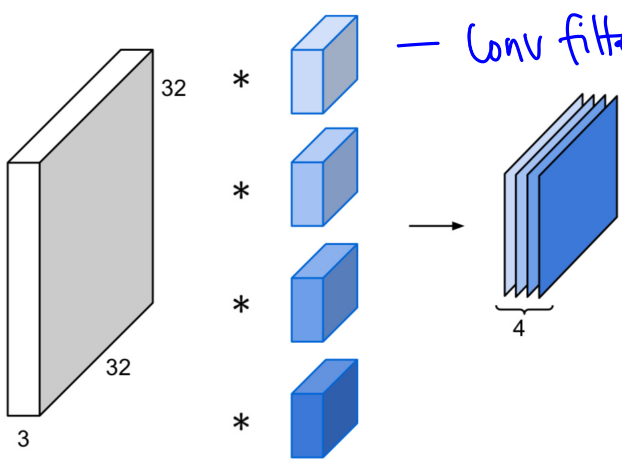
∇ ∇

Allows for gradient shortcircuit

III. CONVOLUTIONAL NETWORKS -

— Conv filter (5x5x3)

image (32x32x3) ←



TODAY: LANGUAGE! (is hard, but I might be biased!)

sense of context becomes extremely relevant!

Q. what are the atomic units, the "pixels" of language?

"Berry the llama was spotted in Baker Lab"

1. words = Berry, the, llama, ... — the underlying dictionary / vocabulary needs to be constantly updated!

2. characters = a, b, c, ... — scalable from a language perspective

10 words = 10T units of compute
10 byte/chars = 40T units of compute!

3. Subwords = "walk", "ing", "chat", "g", "p", "t"

The atomic units are called TOKENS!

Vector representations (+ notion of similarity)

Tokenization : where | are | you | going | ?

Need to represent in a numerical format, more precisely, we need a vector

Iradhi says – use some form of an encoding

$V = \{$ where, when, are, you, going, from, moving, ? $\}$

$|V| = 8$

	where	when	are	you	going	from	moving	?
"where are you going?"	1	0	1	1	1	0	0	1
"when are you moving?"	0	1	1	1	0	0	1	1
"where are you from?"	1	0	1	1	0	1	0	1

DOT - PRODUCT SIMILARITY –

$dot(s^{(1)}, s^{(3)}) = 4$, $dot(s^{(1)}, s^{(2)}) = 3$ ← "matt"

FLAW – SENTENCE STRUCTURE IS SOMEWHAT CAPTURE, BUT SEMANTICS ARE MISSED!

If $|V| = 1m$, each vector is 1M, and sparse (= mostly zeros!)

Capturing "semantics" — a stupid example!

$s^{(1)}$: this building is made of stone

$s^{(2)}$: Ezra Cornell sat in Gates Hall

$s^{(3)}$: this illu is made of stone

$s^{(4)}$: Tushaar went to his illu

"ఇల్లు" — house

Matt said, " ... "

SIMILAR $\underbrace{\text{WORDS}}_{\text{I}}$ APPEAR IN SIMILAR $\underbrace{\text{CONTEXTS}}_{\text{II}}$!

" $\underbrace{\text{this}}$ $\underbrace{\text{illu}}_{\text{word}}$ $\underbrace{\text{is beautiful}}$ "

Context

target token

$$S = (t_1, \ldots, \underbrace{t_{i-w}, \ldots, t_{i-1}}_{\text{context}}, t_i, \underbrace{t_{i+1}, \ldots, t_{i+w}}_{\text{context}}, \ldots)$$

Binary classification that we DON'T care about!

Given a target token, t_i

context tokens, $\{t_{i-w}, \ldots, t_{i-1}, t_{i+1}, \ldots, t_{i+w}\} = C_i$

we want to build a binary cls,

$$P(+1 \mid t_i, C_i) \leftarrow$$ probability that C_i is a true context of t_i

in our e.g., - "illa" = target, C_i = {this, is, beautiful} is the true context.

"Similar" tokens appear in similar ctxs

↓

We can use the notion of some similarity b/w t_i, C_i

$$P(+1 \mid t_i, C_i) = \prod_{-w \le k \le w} P(+1 \mid t_i, t_{i+k})$$

What is the proba that t_{i+k} is a true context token of t_i = "illa"?

ensures normalization b/w 0,1

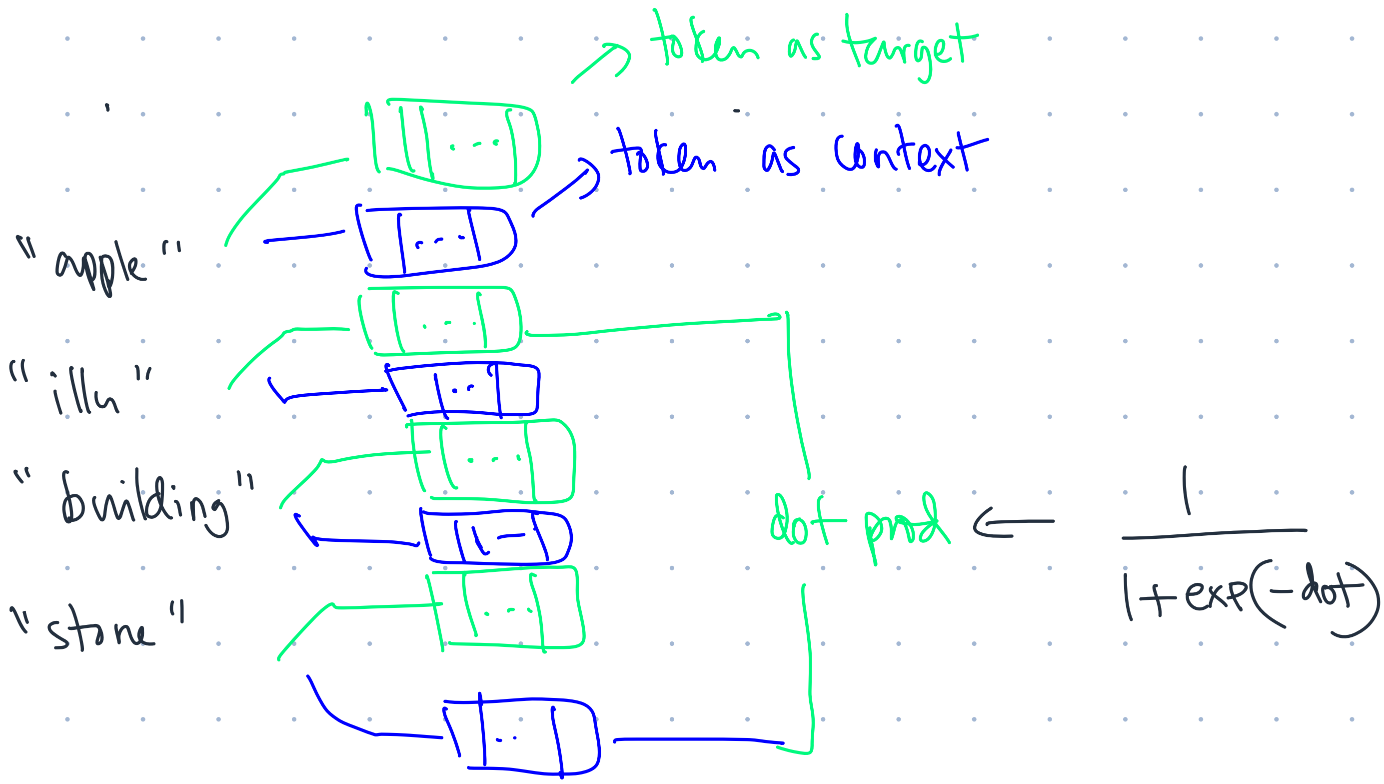
$$= \frac{1}{1 + \exp(-\mathrm{dot}(t_i, t_{i+k}))}$$

Computes similarity b/w target, ctx token

(Similar to logistic regression!)

Q. What should t_i, t_{i+k} be?

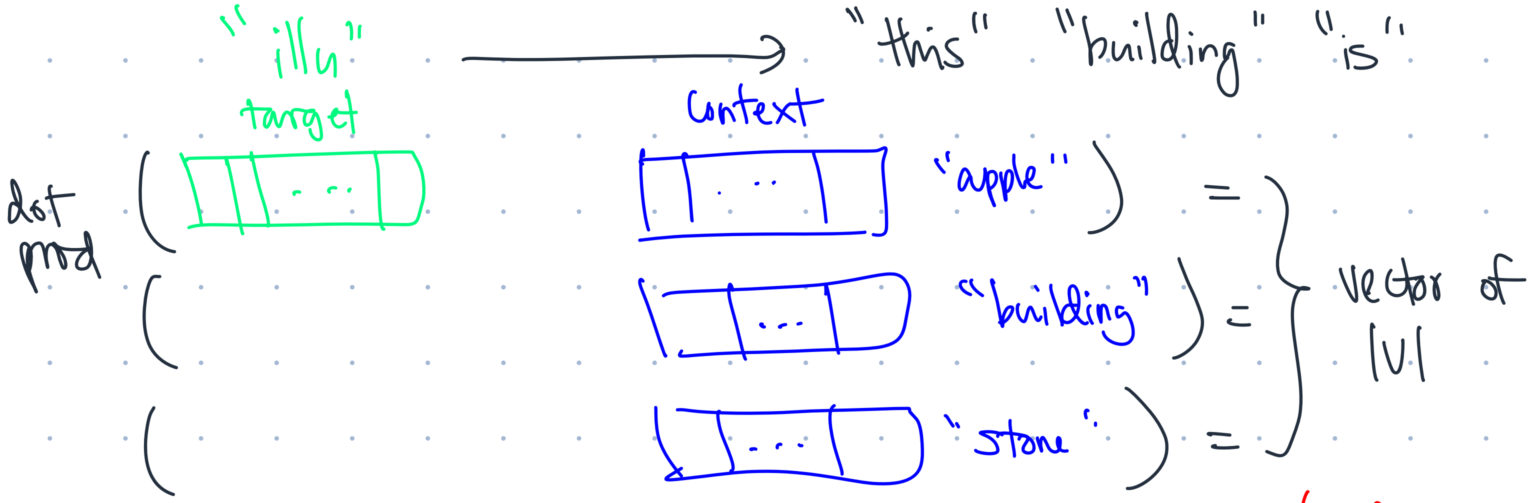
IDK what t_i, t_{i+k} are, but ~~I~~ we can learn them using GD!



OBJ: maximize $P(y \mid \text{true target}, \text{true context})$

— Started the idea of learnable embeddings!

<u>Reformulation</u> – Given a target token, can you predict the context tokens?



Normalization is done using <u>softmax</u> –

$$\frac{\exp(z_j)}{\sum_{k=1}^{n} \exp(z_k)}$$

Aside: PERPLEXITY

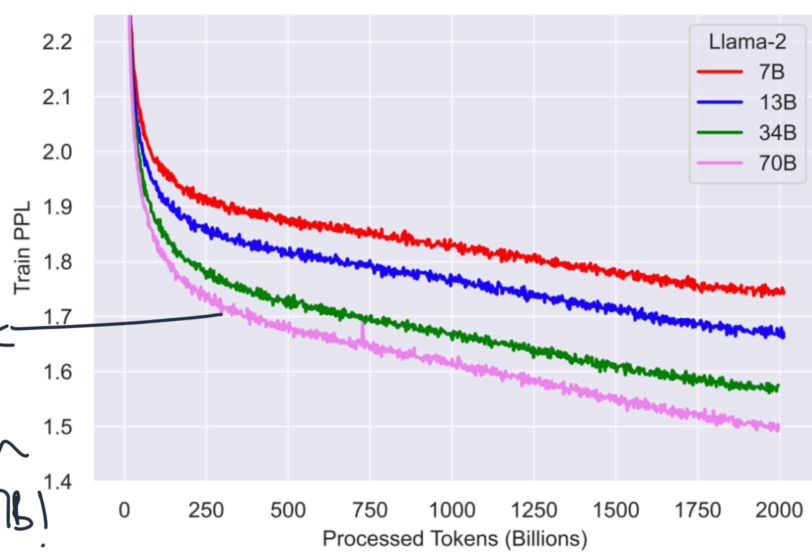
min. NLL loss — straightforward max log-likelihood
take negative → NLL

PERPLEXITY — "Complete this ?"
→ model predicts "sentence"
true is "confirmation"
RESULT — model is shocked!

perplexity
(lower is better)

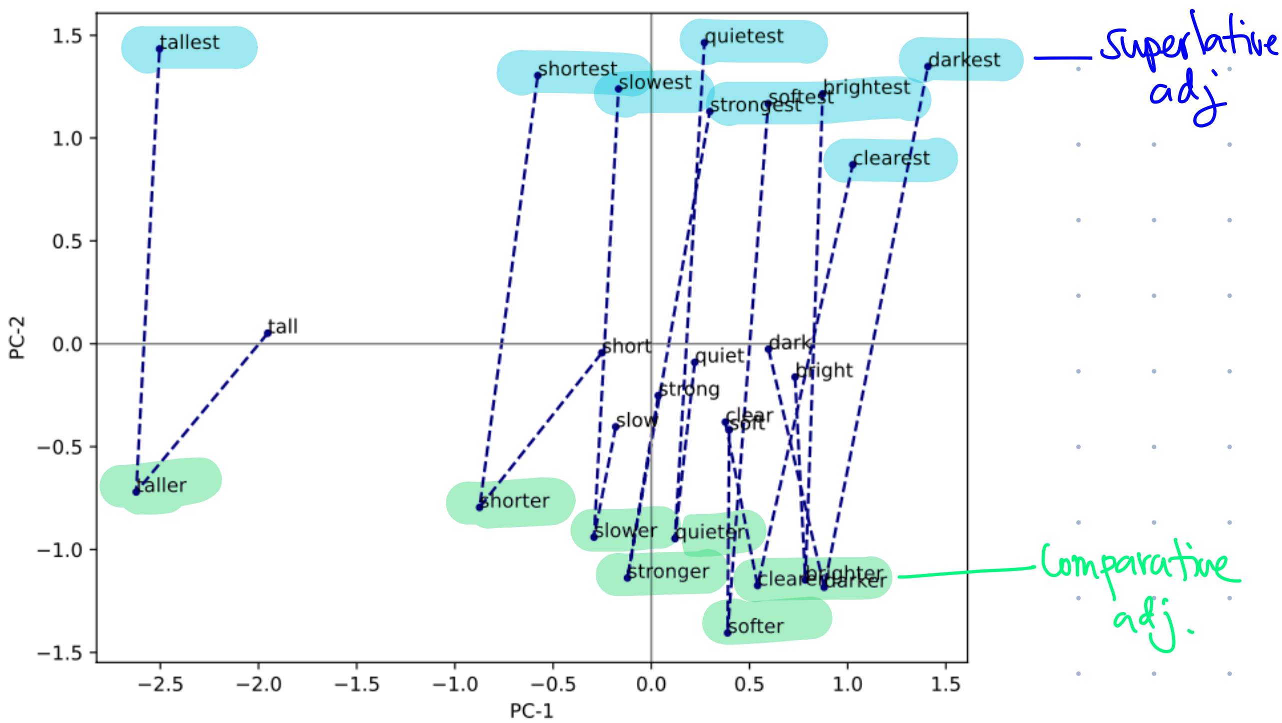
Hyperparams				Dev Set Accuracy		
#L	#H	#A	LM (ppl)	MNLI-m	MRPC	SST-2
3	768	12	5.84	77.9	79.8	88.4
6	768	3	5.24	80.6	82.2	90.7
6	768	12	4.68	81.9	84.8	91.3
12	768	12	3.99	84.4	86.7	92.9
12	1024	16	3.54	85.7	86.9	93.3
24	1024	16	3.23	86.6	87.8	93.7

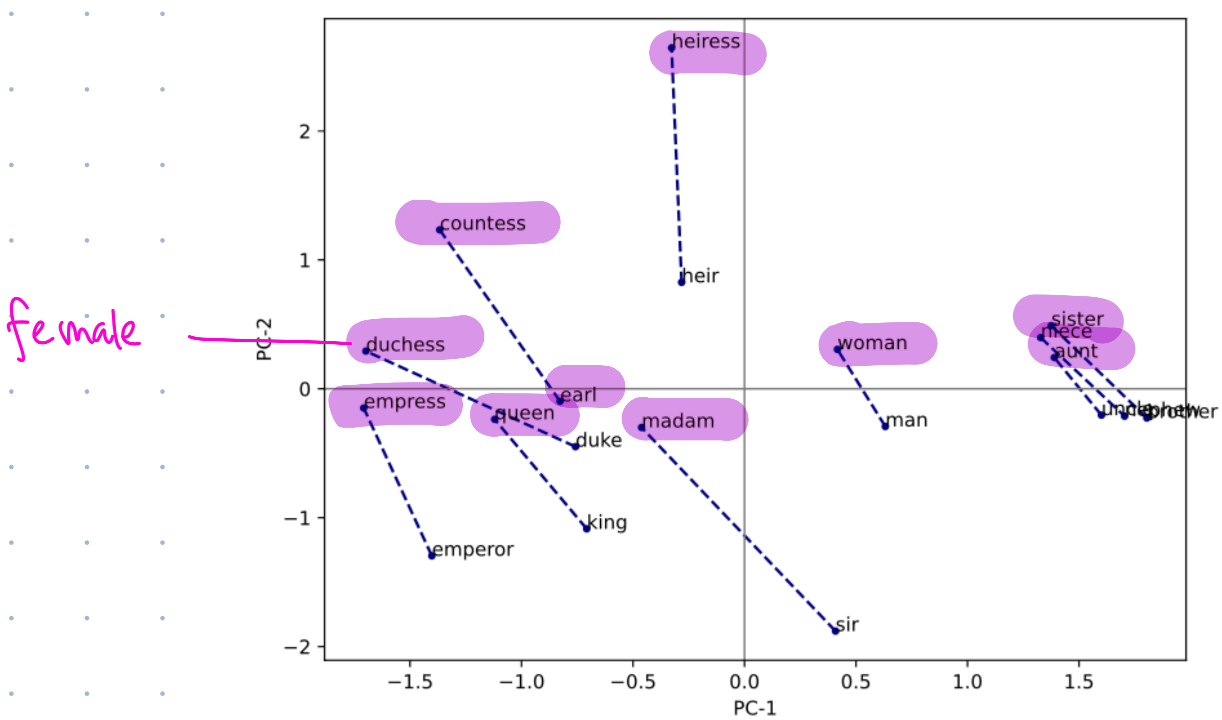
Random NLP Benchmarks
(higher is better)



Llama-70B is known to be much better than Llama-7B!

Visualizing learned embeddings using PCA





$$\overrightarrow{king} - \overrightarrow{man} + \overrightarrow{woman} = \overrightarrow{queen}$$

Static → Dynamic embeddings

Once trained, there learnable embedding are "static"
⇔ One embedding for each token

" The bank teller " vs. " the river bank "
→ same embedding representation

NEED DYNAMIC EMBEDDINGS!

$\boxed{\text{CS 3780 Reviews}}$ → 1 CS 3780 website
(feeling lucky!) RMP
Reddit
⋮

} Ranked

Final document comes
80% from 1 + 20% from 2 + 30% from 3

(softmax) self-attention -

"the" "bank" "teller"

interested in emb for "bank"

dot(vec("the"), vec("bank")) ← dot-prod similarity b/w "the", "bank"
dot(vec("bank"), vec("bank"))
dot(vec("teller"), vec("bank"))

Returned links/ "keys"

search "query"

use them

to form a weighted average of all embeddings

<"the", "bank"> . vec("the") +
<"bank", "bank"> . vec("bank") +
<"teller", "bank"> . vec("teller")

POSITIONAL INFORMATION

bank: "the bank teller" different from bank: "the river bank"

" the bank teller sat at the river bank "
(1) (2) (3) (4) (5) (6) (7) (8)

Attention computations + Efficiency

$$O = \text{Softmax}\left(QK^T\right)V$$

X is a "n" long sequence, with d-dimensional tokens

Time - $O(n^2 d)$
Space - $O(n^2)$ to store QK^T

$X \in \mathbb{R}^{n \times d}$

$Q \quad K \quad V \in \mathbb{R}^{n \times d}$

linear(x) linear(x) linear(X)