# Machine Learning

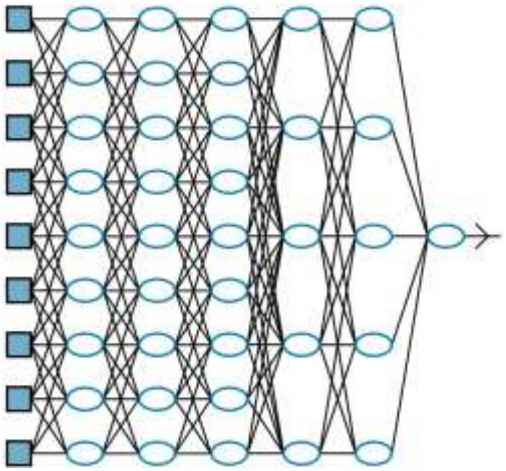
week 4, 2024

# Review

**Neural networks**: layers of units/artificial neurons

**Weights** control how information from inputs influences the output layer

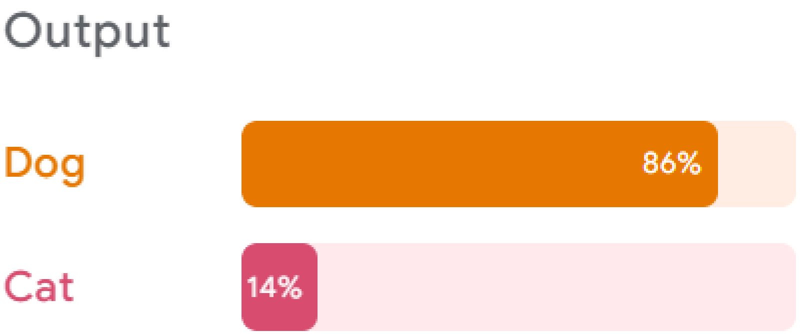
TODO: sync to week 2/3 content



# Overview

- probability  (softmax)
- optimization (loss)
- representation of data (images)
- representation of data (text)
- learning an embedding
- similarity and distance of embeddings

# Probability in Machine Learning




What does the output say?

1. “Dog”, with probability 0.86
2. It is certainly a cat or a dog, probability 1.0

Discuss: is this useful? Is this correct?

# Probability in Machine Learning

$x$ – input, such as 

$y$ – output, "Dog"

$P(y|x)$ - the neural network model

Read "probability of $y$, given $x$"

# Probability in Machine Learning

$P(y|x)$ - the neural network model

How to make this happen?

**Step 1**: make the output *look* like probabilities using softmax

$$softmax(y_k) = \frac{e^{y_k}}{\sum_{k'}^{d} e^{y_{k'}}}$$

$d$ – number of outputs

| $y_k$ | $softmax(y_k)$ | class |
|---|---|---|
| -0.12 | 0.107 | cat |
| 2 | 0.893 | dog |

# Probability in Machine Learning

$P(y|x)$ - the neural network model

How to make this happen?

**Step 2**: train the network to match the real probability

$x =$ 

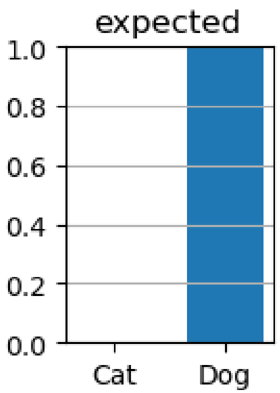
$P(y = \text{Cat}|x)$ ?

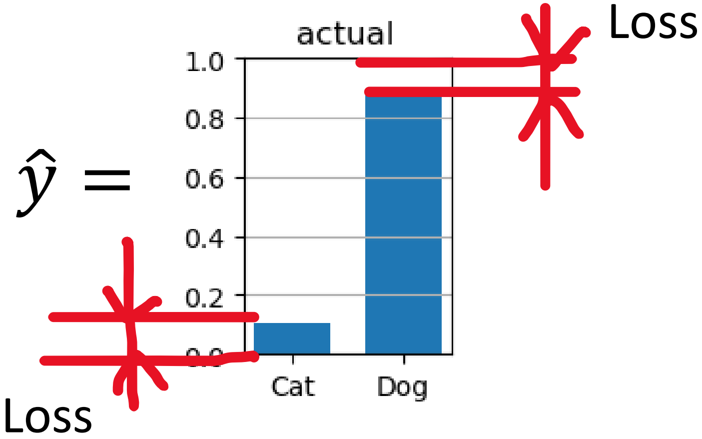
$P(y = \text{Dog}|x)$ ?

what should
these be?

# Training the Network

$x =$ 

$y =$ 


$\hat{y} =$ 


Loss measures how “wrong” the output is

Minimize this:

$$L = -y_{Cat} \log \hat{y}_{Cat} - y_{Dog} \log \hat{y}_{Dog}$$

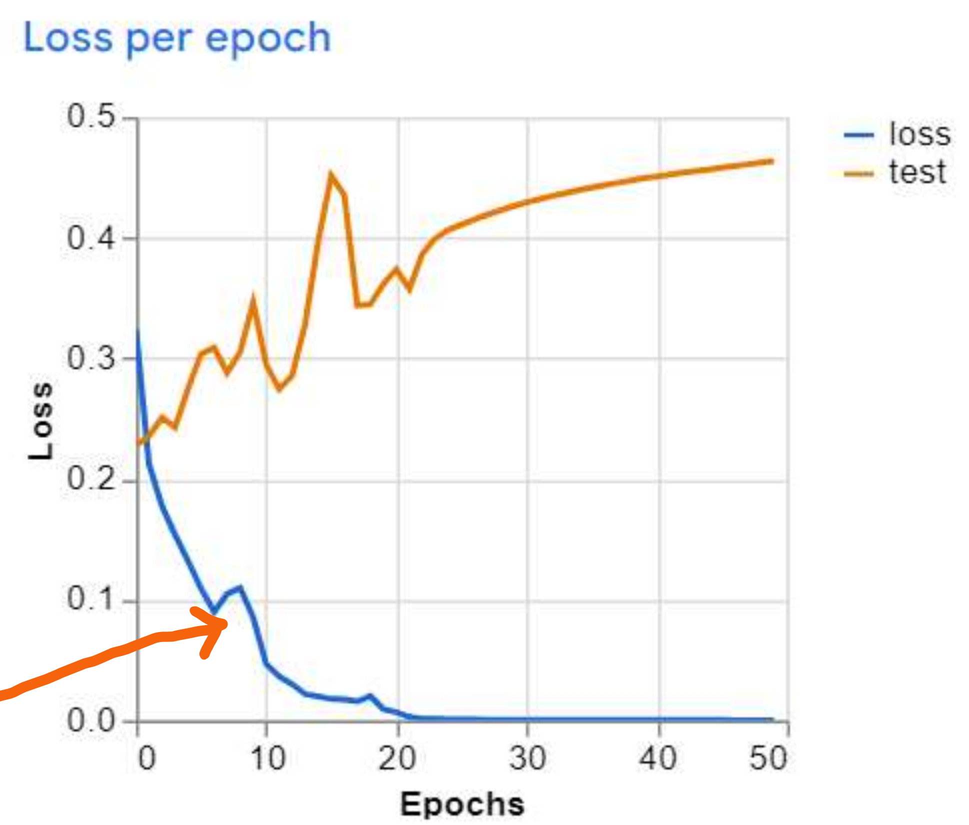
# Training the Network

$$L = -y_{Cat} \log \hat{y}_{Cat} - y_{Dog} \log \hat{y}_{Dog}$$

can change

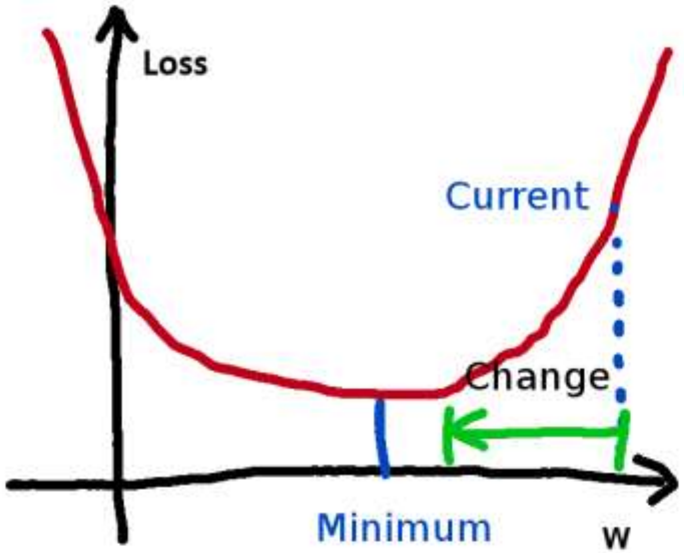
Discuss: *how* can you change them?

successful training:
loss decreases

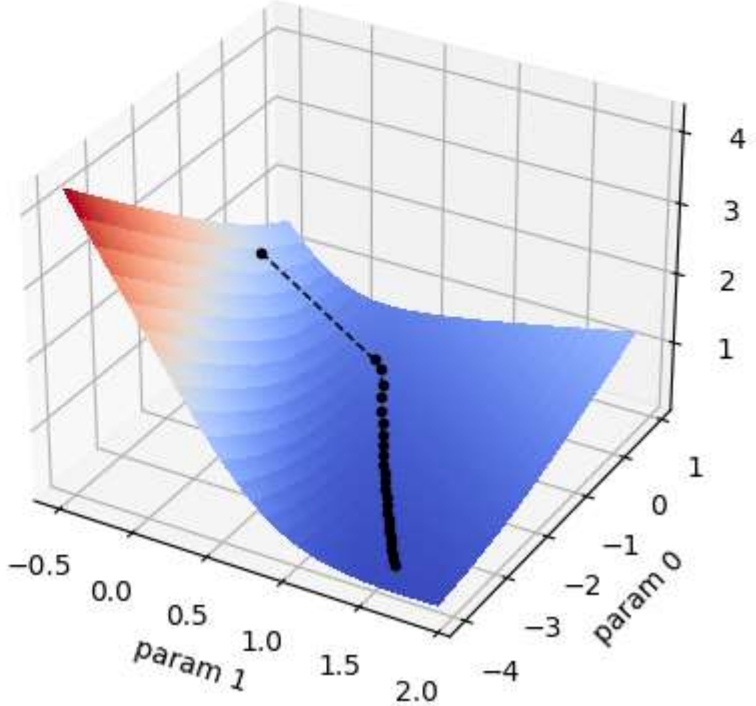


# Training the Network

2 param network,
classify *I. Setosa*
by petal length




"downhill" defined by the **gradient**




Change the parameters in "downhill" direction on loss surface

# Representing Images

CIFAR-10 image of a horse (32x32)

Color channels:

32 numbers

```
[[ 39  44  47  72  72  46  50  63  71  74  98  85  68  88  79  60  73 100
   92  73  62  66  63  48  57  52  47  44  41  45  53  47]
 [ 38  41  39  55  66  48  59  66  61  71 117 102  67  76 103 119 126 133
  127 120 107 111  90  61  75  78  72  80  86  97 101 100]
```

total 32 rows ...

```
 [123 124 124 114 120 117 117 127 134 131 127 124 121 123 120 118 111 119
  124 102 118 117  89  83 107 110  97 113 117 100  99  96]]
```

blue channel

# Representing Images

You see:

Neural network sees:  32x32x3 = 3072 numbers

```
[ 28  30  33  62  63  31  29  42  55  67  92  76  57  75  69  57  74  98
  86  71  59  62  57  42  51  46  41  38  37  43  52  46  27  27  21  38
  60  39  41  47  48  72 120 103  66  75 110 134 146 153 146 139 125 130
 109  76  82  85  82  91  99 112 117 115  34  33  24  38  82  52  43  55
 123 130 140 123 128 122 110 103 103 109 114 128 120 119 119 111 123 124
 124 114 120 117 117 127 134 131 127 124 121 123 120 118 111 119 124 102
 118 117  89  83 107 110  97 113 117 100  99  96]
```

# Representing Text

**Step 1**:  words to numbers

*"A blackbird is a black bird"*

| 32 | 2042 | 16944 | 318 | 257 | 2042 | 6512 |
|---|---|---|---|---|---|---|
| 'A' | ' black' | 'bird' | ' is' | ' a' | ' black' | ' bird' |

("tokenized" using tiktoken, GPT-2 encoding)

# Representing Text

| | | | | | | |
|---|---|---|---|---|---|---|
| 32 | 2042 | 16944 | 318 | 257 | 2042 | 6512 |
| 'A' | ' black' | 'bird' | ' is' | ' a' | ' black' | ' bird' |

Problem solved? No.

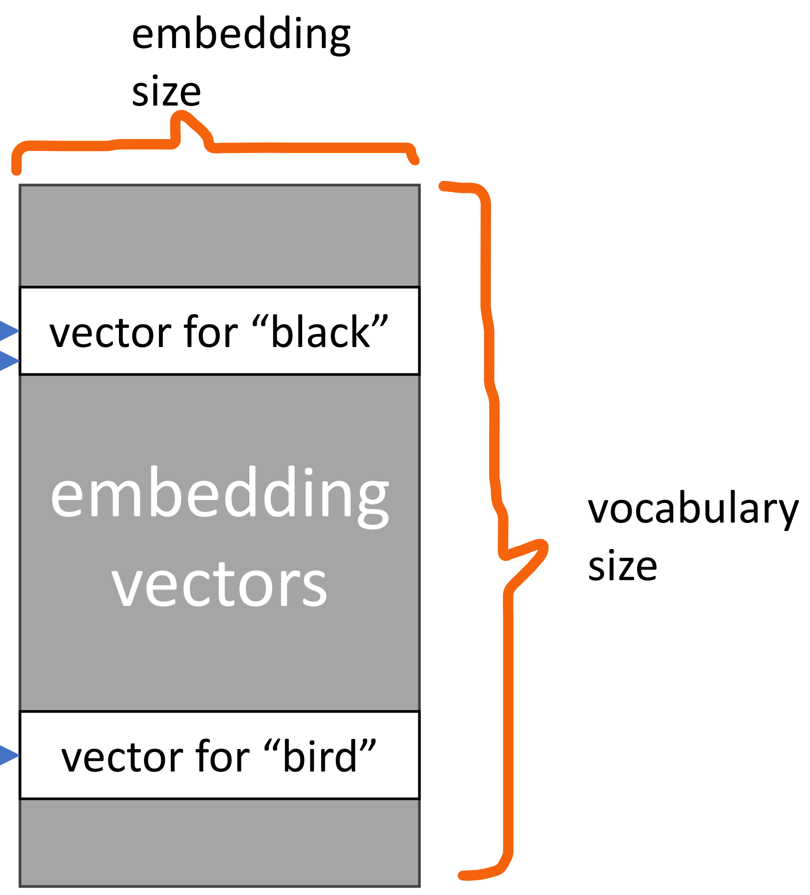
Similar vectors (add +1):

| | | | | | | |
|---|---|---|---|---|---|---|
| 33 | 2043 | 16945 | 319 | 258 | 2043 | 6513 |
| 'B' | 'IT' | '133' | ' on' | 'he' | 'IT' | 'fo' |

# Representing Text

**Step 2**: learn embeddings

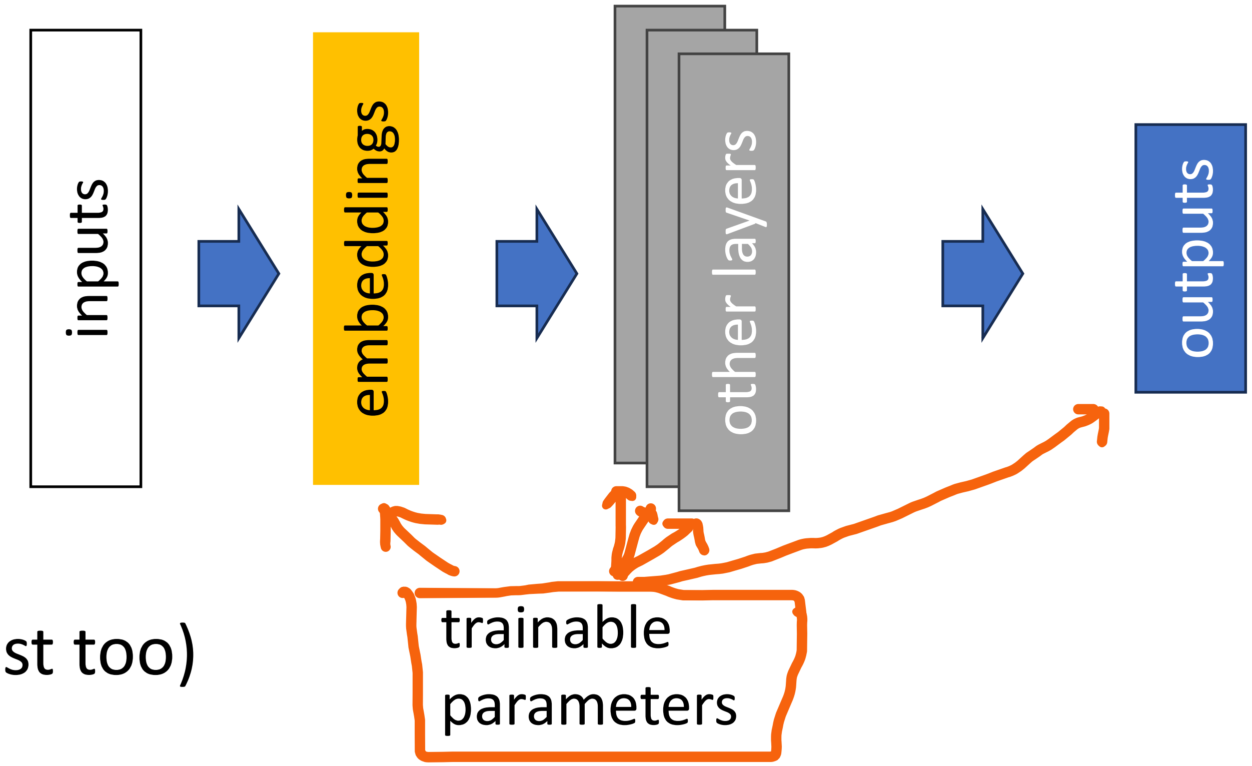
| | |
|---|---|
| 'A' | 32 |
| ' black' | 2042 |
| 'bird' | 16944 |
| ' is' | 318 |
| ' a' | 257 |
| ' black' | 2042 |
| ' bird' | 6512 |




Embedding sizes: 150, 512, 3072, …

# Representing Text

Can learn embeddings during network training:




(Pre-trained embeddings exist too)

# Embeddings and Meaning

What is an “embedding”?

GloVe pre-trained embedding for “*bird*”, size 200:

```
array([ 0.5612   , -0.92374  , -0.73493  , -0.47596  ,  0.12066  ,
       -0.35696  , -0.66272  , -0.27035  , -0.76995  , -0.15108  ,
       -0.23904  ,  0.15406  , -0.0064953, -0.025873 , -0.055705
       -0.35045  ,  0.2271   ,  0.49246  ,  0.42676  ,  0.070232 ,
       -0.1377   , -0.1964   ,  0.14237  ,  0.5167   , -0.52172  ,
        0.10113  , -0.14689  , -0.027673 , -0.42438  , -0.3572   ]
```

# Embeddings and Meaning

GloVe embeddings,

general purpose

2D projection of words

seen in e.g. movie reviews

negative

verbs?

positive