CPSC 340:
Machine Learning and Data Mining

Discrete Labels

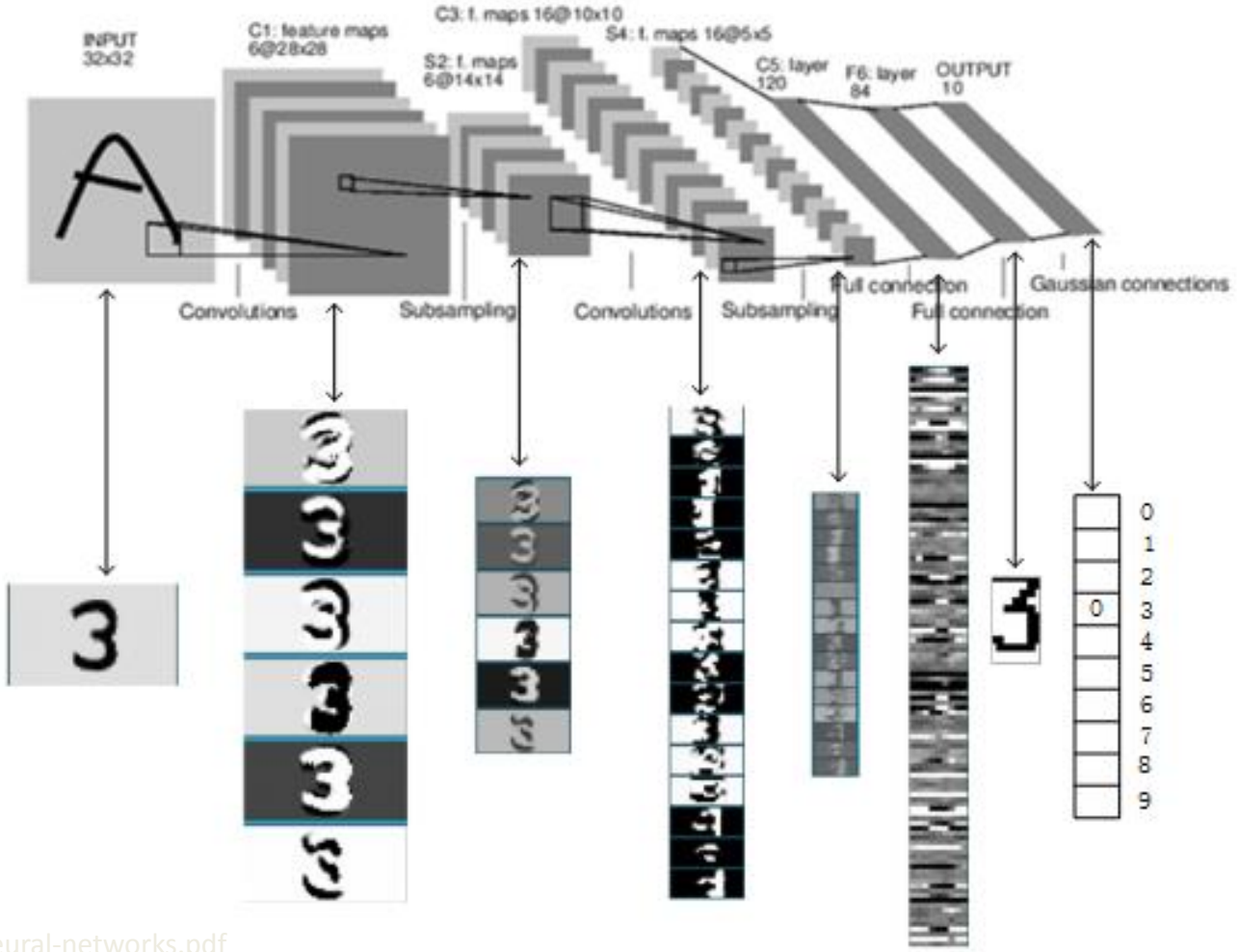
Fall 2015

Admin

- Assignment 5 is posted.
  - Due Friday of next week.
  - A2.2 update: use k = 10.

Last Time: Convolutional Neural Networks

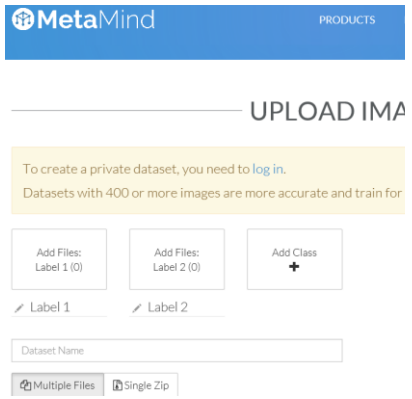
- Convolutional neural networks:
  1. Convolutional layers.
  2. Pooling layers.
  3. Fully-connected layers.

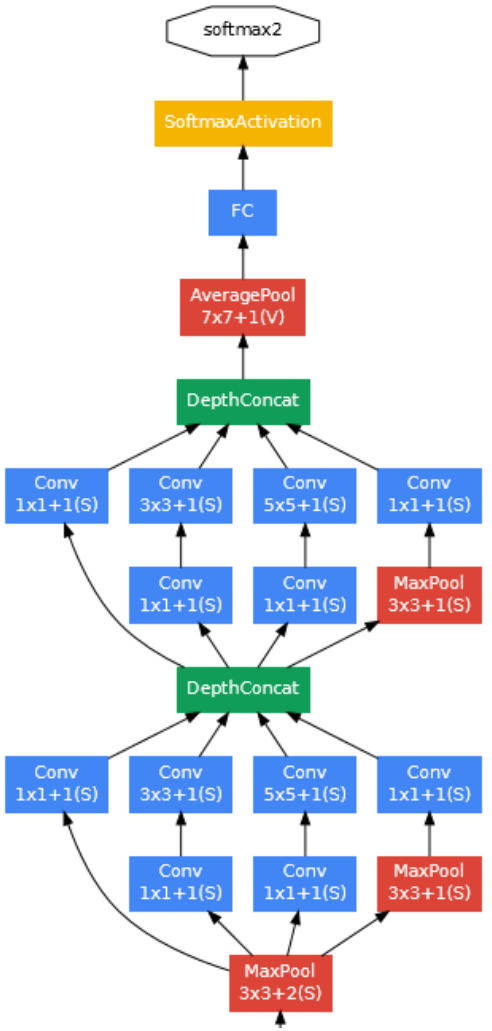


http://blog.csdn.net/strint/article/details/44163869
http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

GoogLeNet

- GoogLeNet is very deep competitive-winning system.
- Training these systems is very expensive:
  - Weeks on clusters of computers with expensive GPUs.
- What if you aren't Google?
  - Some groups release their features/network code.
  - Quickly learn to identify new objects by:
    - Use features learned from millions of images and thousands classes.
    - Or 'fine-tune' the entire network with your data.





http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf
https://www.metamind.io/vision/train

Today: Alternatives to Squared Loss

- We've been using squared error as our default 'loss':

$$f(\hat{y}_i) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- A lot of models we've discussed fit in this framework:

Linear regression:

$$\hat{y}_i = w^T x_i$$

PCA:

$$\hat{y}_{ij} = w_j^T z_i$$

Collaborative fitering:

$$\hat{y}_{um} = b_u + b_m + w_m^T z_u$$

Deep neural networks:

$$\hat{y}_i = w^T h(W_{(3)} h(W_{(2)} h(W_{(1)} x_i)))$$

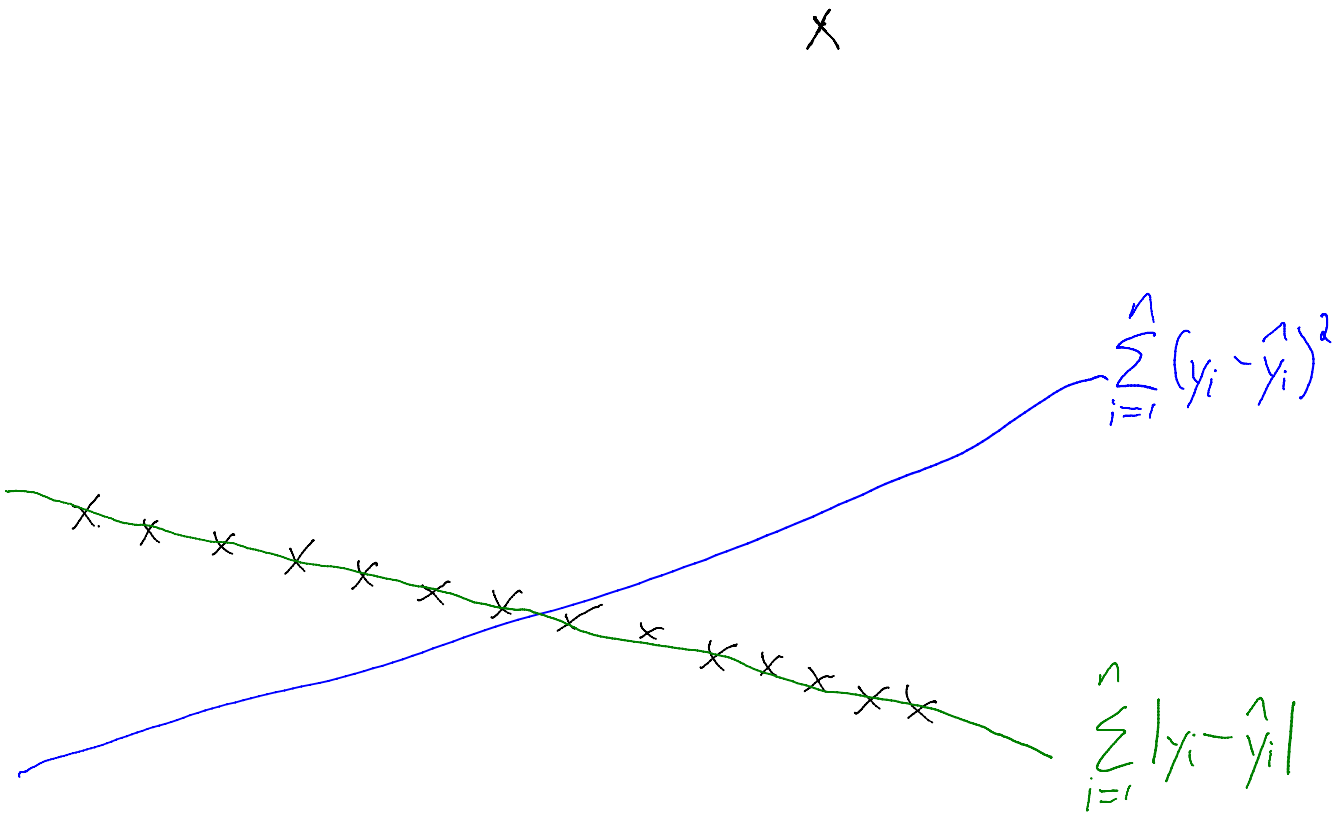
- Square error is differentiable and sometimes has closed-form.
- But, usually squared error is not the right 'loss'.
  - Today we discuss alternatives.

Review: Robust, Brittle, and Logistic Regression

- We previously discussed a few alternatives:
  - L1-error and Huber loss are more robust to outliers.

$$\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

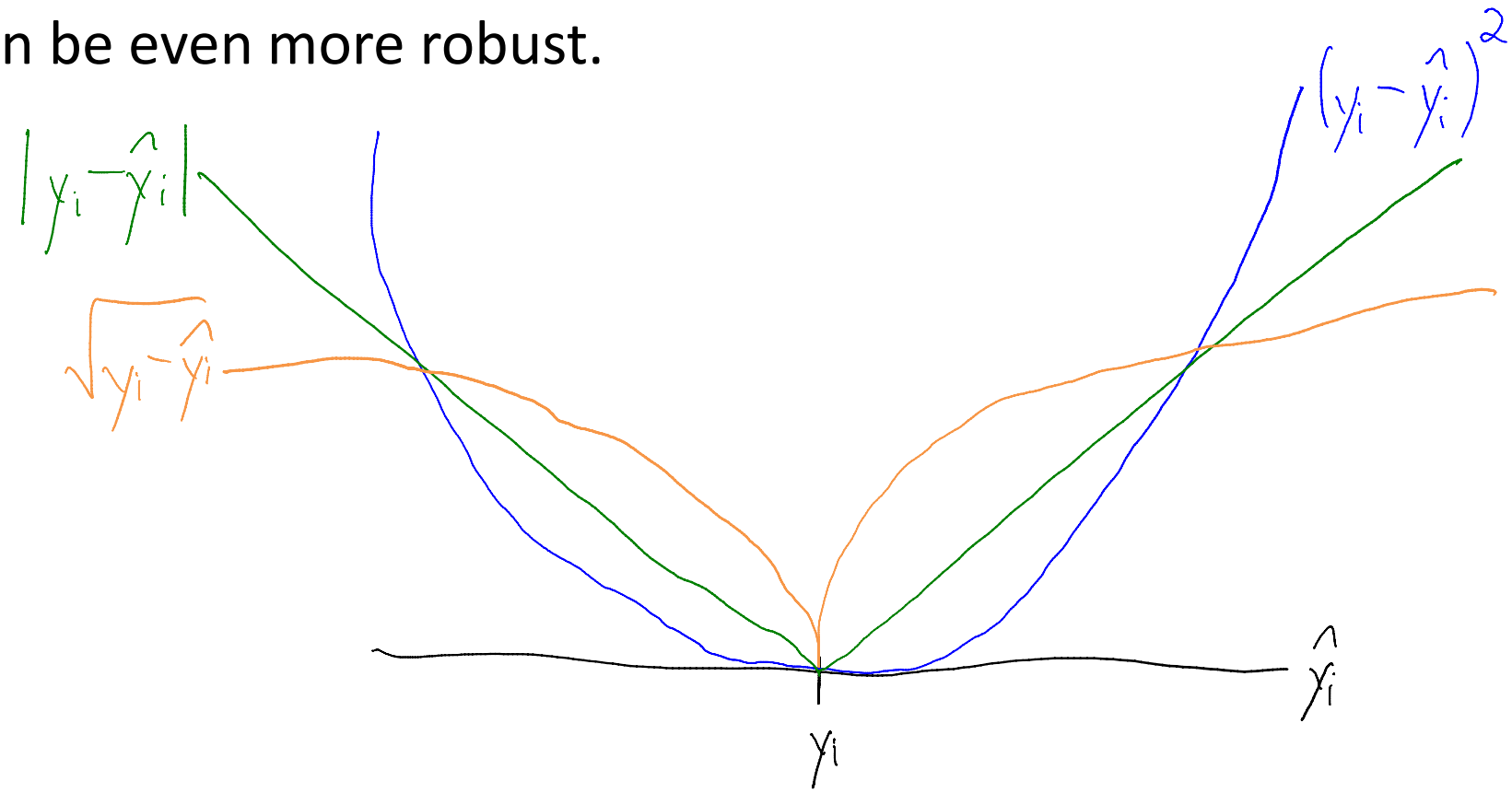
Use this instead of squared to make PCA/recommender/neural-network robust to outliers.



Review: Robust, Brittle, and Logistic Regression

- We previously discussed a few alternatives:
  - L1-error and Huber loss are more robust to outliers.
  - Non-convex losses can be even more robust.

$$\sum_{i=1}^{n} \sqrt{|y_i - \hat{y}_i|}$$

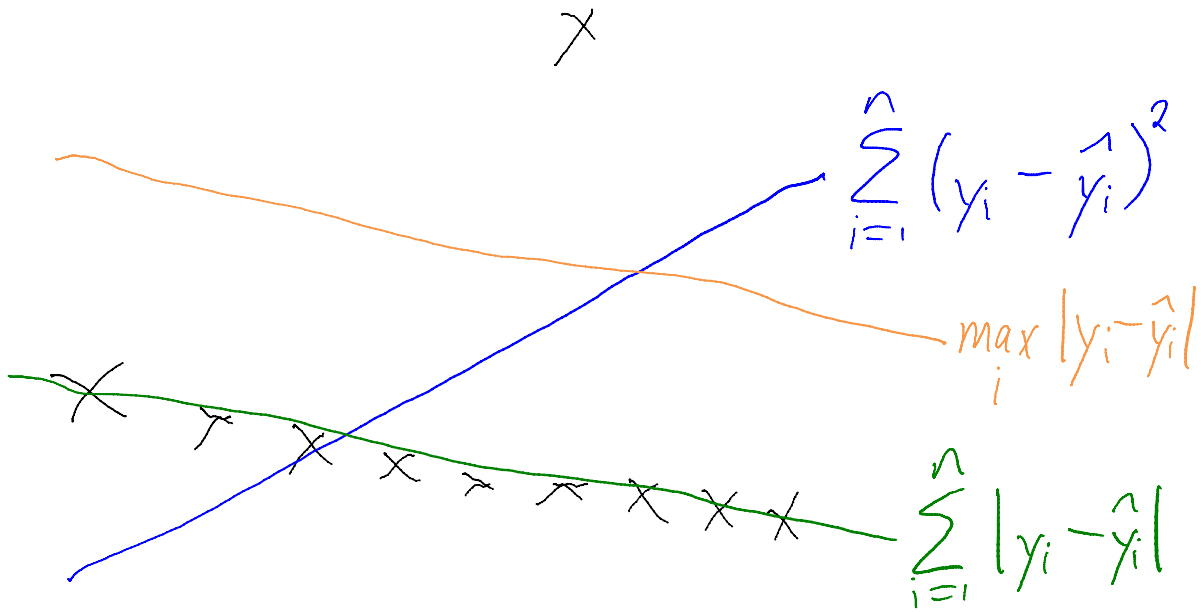


Review: Robust, Brittle, and Logistic Regression

- We previously discussed a few alternatives:
  - L1-error and Huber loss are more robust to outliers.
  - Non-convex losses can be even more robust.
  - Maximum loss gives better performance in worst case.

$\max_i |y_i - \hat{y}_i|$

use this to make neural network focus on worst case.

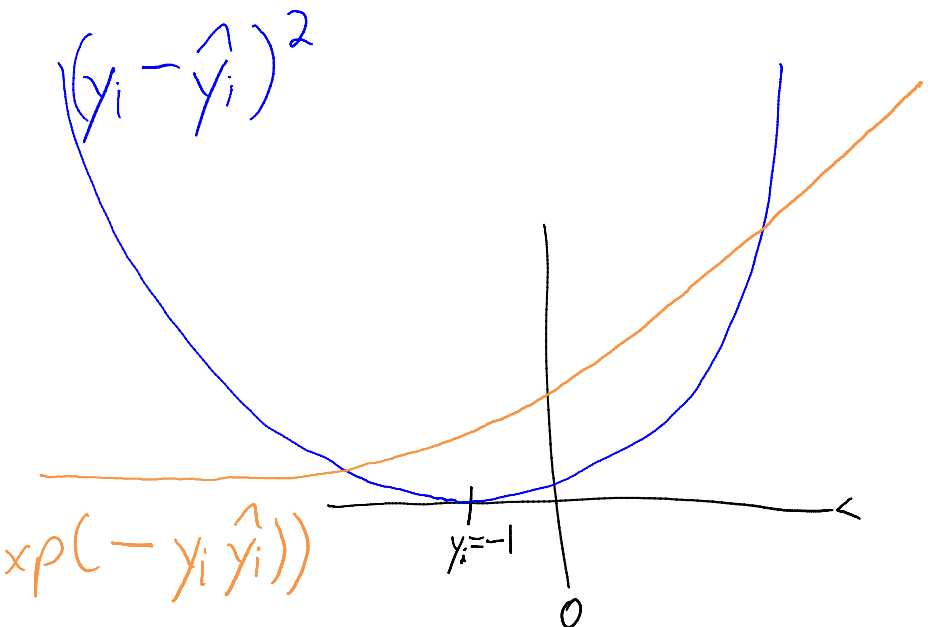


Review: Robust, Brittle, and Logistic Regression

- We previously discussed a few alternatives:
  - L1-error and Huber loss are more robust to outliers.
  - Non-convex losses can be even more robust.
  - Maximum loss gives better performance in worst case.
  - Hinge and logistic losses are better for binary data.

$$\sum_{i=1}^{n} \log(1+\exp(-y_i \hat{y}_i))$$

→ use this for PCA with binary data, or neural network for binary classification



Review: Robust, Brittle, and Logistic Regression

- We previously discussed a few alternatives:
  - L1-error and Huber loss are more robust to outliers.
  - Non-convex losses can be even more robust.
  - Maximum loss gives better performance in worst case.
  - Hinge and logistic losses are better for binary data.
- What about other types of discrete labels?
  - Multi-label: {'cat', 'dog', 'human'}.
  - Categorical: {'Edmonton', 'Paris', 'Philadelphia', 'Vancouver'}.
  - Ordinal: {1 star, 2 stars, 3 stars, 4 stars, 5 stars}.
  - Counts: 602 'likes'.
  - Ranking: Difficulty(A3) > Difficulty(A4) > Difficulty (A2) > DifficultyA(1).

Probabilistic Models and Loss Functions

- We can use probabilistic models to derive loss functions.
- Main idea:
  1. Define probability of each possible label: $p(y_i \mid \hat{y}_i)$ $\hat{y}_i = w^T x_i$
  2. Define loss as negative logarithm of the probability. $-\log p(y_i \mid \hat{y}_i)$
- Why???
  - We want predictions that maximize the probability of the label y_i.
  - Taking logarithm doesn't change location of maximum.
  - Maximizing logarithm is equivalent to minimizing negative of logarithm.

$$\arg\max_{\hat{y}_i} p(y_i \mid \hat{y}_i) \iff \arg\max_{\hat{y}_i} \log(p(y_i \mid \hat{y}_i)) \iff \arg\min_{\hat{y}_i} -\log(p(y_i \mid \hat{y}_i))$$

Sigmoid Probabilities and Logistic Loss

- Example of going from probabilities to loss function (binary y_i):

Define $p(y_i = +1 | \hat{y}_i) = \frac{1}{1+\exp(-\hat{y}_i)}$

sigmoid function

Ensures that $p(y_i = +1 | \hat{y}_i) \in (0, 1)$

We need $p(y_i = +1 | \hat{y}_i) + p(y_i = -1 | \hat{y}_i) = 1$ for this to be a valid probability.

This implies $p(y_i = -1 | \hat{y}_i) = \frac{1}{1+\exp(\hat{y}_i)}$

Can write both cases as $p(y_i | \hat{y}_i) = \frac{1}{1+\exp(-y_i \hat{y}_i)}$

Sigmoid Probabilities and Logistic Loss

- Example of going from probabilities to loss function (binary y_i):

Define $p(y_i = +1 | \hat{y}_i) = \frac{1}{1+\exp(-\hat{y}_i)}$ (sigmoid function)

Ensures that $p(y_i = +1 | \hat{y}_i) \in (0, 1)$

We need $p(y_i = +1 | \hat{y}_i) + p(y_i = -1 | \hat{y}_i) = 1$ for this to be a valid probability.

This implies $p(y_i = -1 | \hat{y}_i) = \frac{1}{1+\exp(\hat{y}_i)}$

Can write both cases as $p(y_i | \hat{y}_i) = \frac{1}{1+\exp(-y_i \hat{y}_i)}$

Take logarithm of probability:

$$\log(p(y_i | \hat{y}_i)) = \log\left(\frac{1}{1+\exp(-y_i \hat{y}_i)}\right) = \log(1) - \log(1+\exp(-y_i \hat{y}_i))$$

Loss is negative logarithm of probability: $-\log(p(y_i | \hat{y}_i)) = \log(1+\exp(-y_i \hat{y}_i))$ → logistic loss

Why Logarithm?

- We want loss function to be additive across examples:

$$\sum_{i=1}^{n} g(y_i, \hat{y}_i) \quad \text{for some function 'g'}$$

- If training examples are IID, probability is multiplicative:

$$p(y_1, y_2, y_3, \ldots, y_n \mid \hat{y}_1, \hat{y}_2, \hat{y}_3, \ldots, \hat{y}_n) = p(y_1 \mid \hat{y}_1)\, p(y_2 \mid \hat{y}_2)\, p(y_3 \mid \hat{y}_3) \cdots p(y_n \mid \hat{y}_n)$$

- Logarithm of the probability is additive across examples:

$$\log(p(y_1, y_2, \ldots, y_n \mid \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n)) = \log(p(y_1 \mid \hat{y}_1)) + \log(p(y_2 \mid \hat{y}_2)) + \cdots + \log(p(y_n \mid \hat{y}_n))$$

Gaussian Probabilities and Squared Loss

Consider a continuous label y_i:

If y_i is a normal/Gaussian distribution with mean $\hat{y}_i$, then

$$p(y_i \mid \hat{y}_i) \propto \exp\left(-\frac{(y_i - \hat{y}_i)^2}{2\sigma^2}\right)$$

↑ "multiplied by factors not depending on y_i"

Take negative of logarithm:

$$-\log(p(y_i \mid \hat{y}_i)) = -\log\left(\exp\left(-\frac{(y_i - \hat{y}_i)^2}{2\sigma^2}\right)\right) - \text{constant}$$

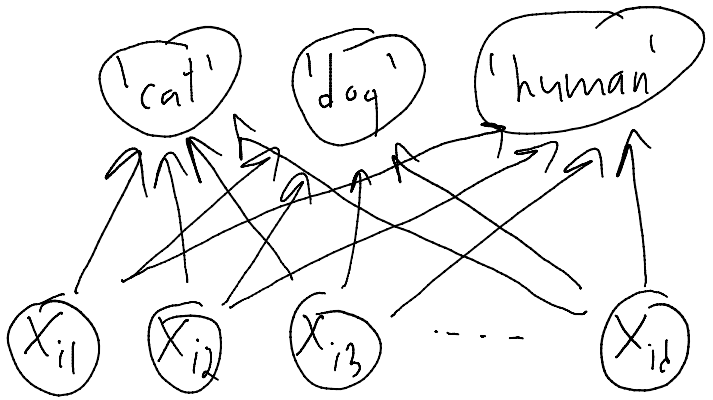
↳ terms not depending on y_i

$$= \frac{1}{2\sigma^2}(y_i - \hat{y}_i)^2 - \text{constant}.$$

Squared error when $\sigma^2 = 1$ and we ignore constant

Multi-Label Data and Independent Logistic Losses

- Consider the case of multi-label data like {'cat', 'dog', 'human'}.
  - Image can have none, some, or all of these labels. E.g., $y_i = \{1,-1,1\}$.
- We can treat this case in a similar way to latent-factor models:



$$\hat{y}_{ij} = w_j^T x_i$$

we have a weight vector 'w_j' for each category 'j'

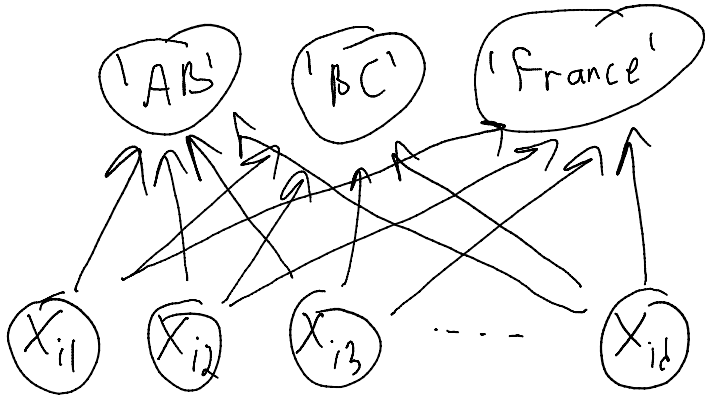
Make predictions using $\text{sign}(w_j^T x_i)$.

- Squared error has 'bad' errors: penalized for being too right.
- We could use logistic loss on each category:

$$\underset{W \in \mathbb{R}^{3\times d}}{\text{argmin}} \sum_{i=1}^{n} \sum_{j=1}^{3} \log(1+\exp(-y_i w_j^T x_i))$$

Categorical Data and One vs. All

- Categorical data: we have multiple labels but only one is correct.
  - Classifying images as taken in {'Alberta', 'Paris', 'Philadelphia', Vancouver'}.
  - In this case, we could have $y_i = 3$.
- We can use same model as previous slide:



$$\hat{y}_{ij} = w_j^T x_i$$

- To make a single prediction, take the biggest $w_j^T x_i$.
  - 'One vs. all' classifier.

$$y_i = \max_j \{ w_j^T x_i \}$$

Categorical Data and Softmax Loss

- Disadvantage of 'one vs. all':
  - Logistic loss focuses on {0,1} decisions, not making $w_j^T x_i$ large for correct y_{ij}.
  - We want a loss that makes $w_j^T x_i$ big for correct y_{ij}, small for others.
- 'Softmax' or multinomial logistic regression model:

$$p(y_i \mid \hat{y}_{i1}, \hat{y}_{i2}, \hat{y}_{i3}, \hat{y}_{i4}) = \frac{\exp(w_{y_i}^T x_i)}{\exp(w_1^T x_i) + \exp(w_2^T x_i) + \exp(w_3^T x_i) + \exp(w_4^T x_i)}$$

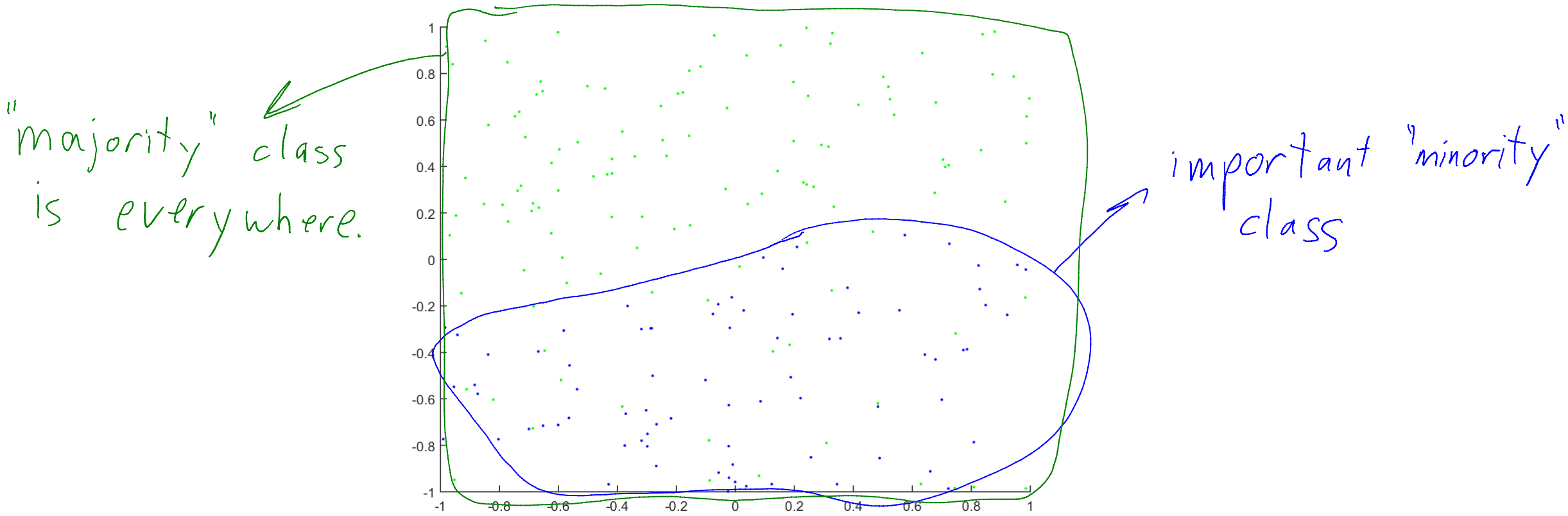
← training label

← all possible labels (makes it a valid probability)

- Exponential magnifies large values: most probability is on max.
- Negative of logarithm of probability gives loss function.
  - *Logistic loss is special case* where $w_1=0$.

Unbalanced Data and Extreme-Value Loss

- Consider binary case where:
  - One class overwhelms the other class ('unbalanced' data).
  - Really important to find the minority class (e.g., minority class is tumor).



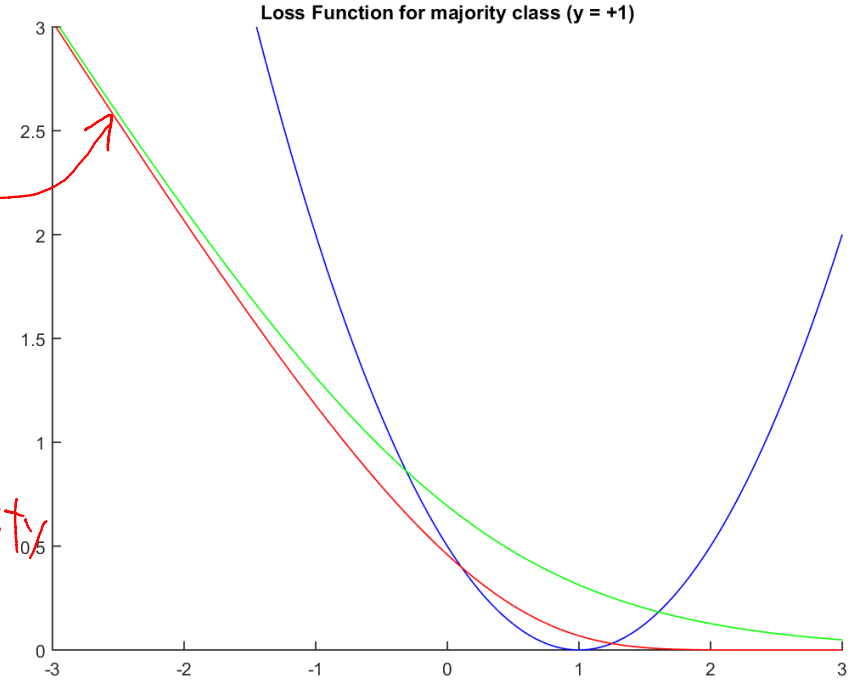
Unbalanced Data and Extreme-Value Loss

- Extreme-value distribution:

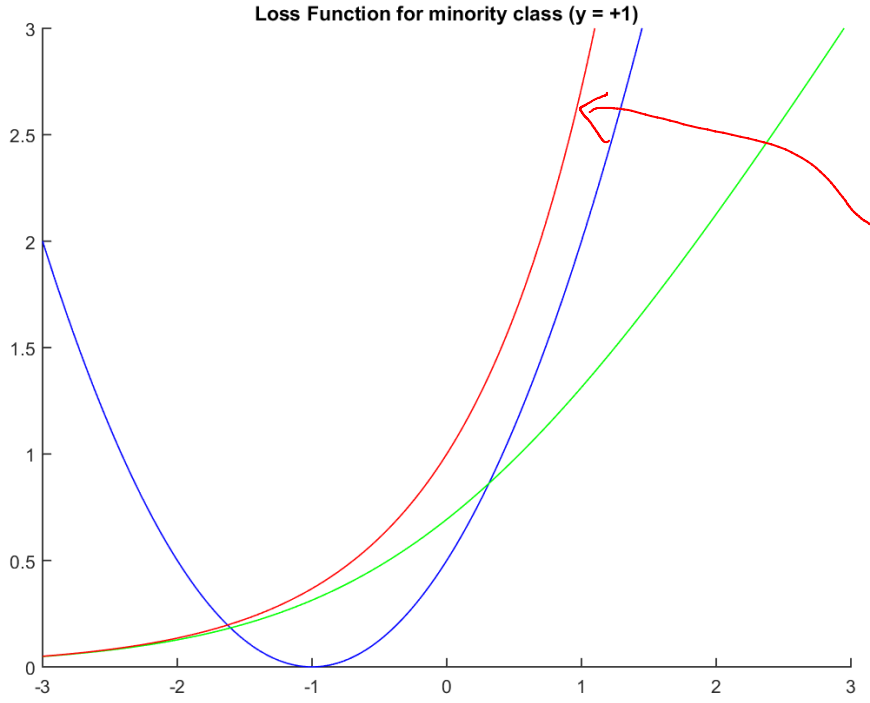
$$p(y_i = +1 \mid \hat{y}_i) = 1 - \exp(-\exp(\hat{y}_i)) \quad [+1 \text{ is majority class}]$$

asymmetric

To make it a probability, $p(y_i = -1 \mid \hat{y}_i) = \exp(-\exp(\hat{y}_i))$



Similar to logistic for majority class



big penalty for getting minority class wrong.

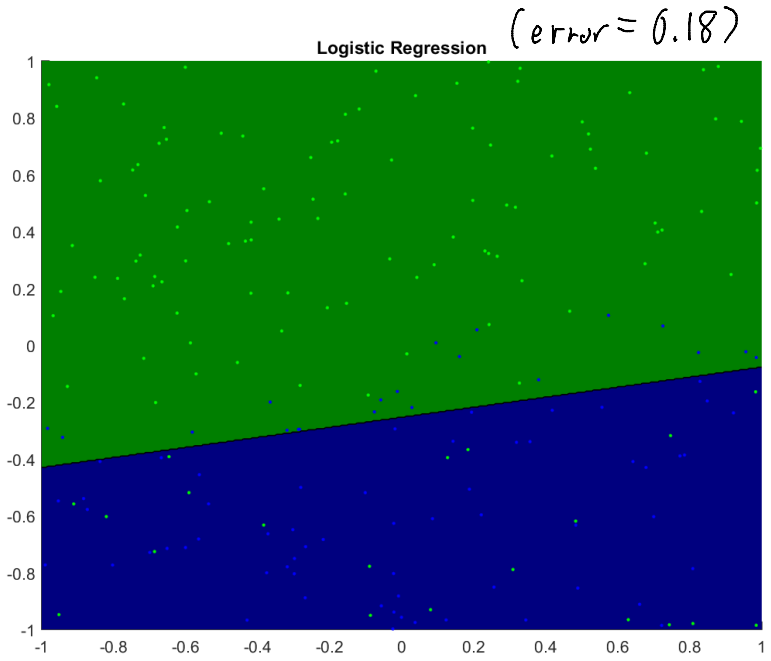
Unbalanced Data and Extreme-Value Loss

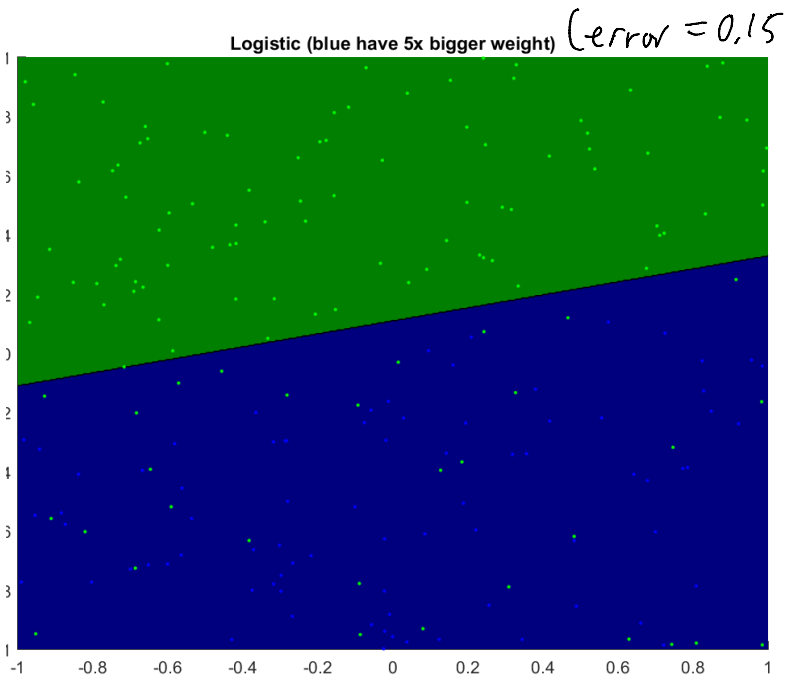
- Extreme-value distribution:

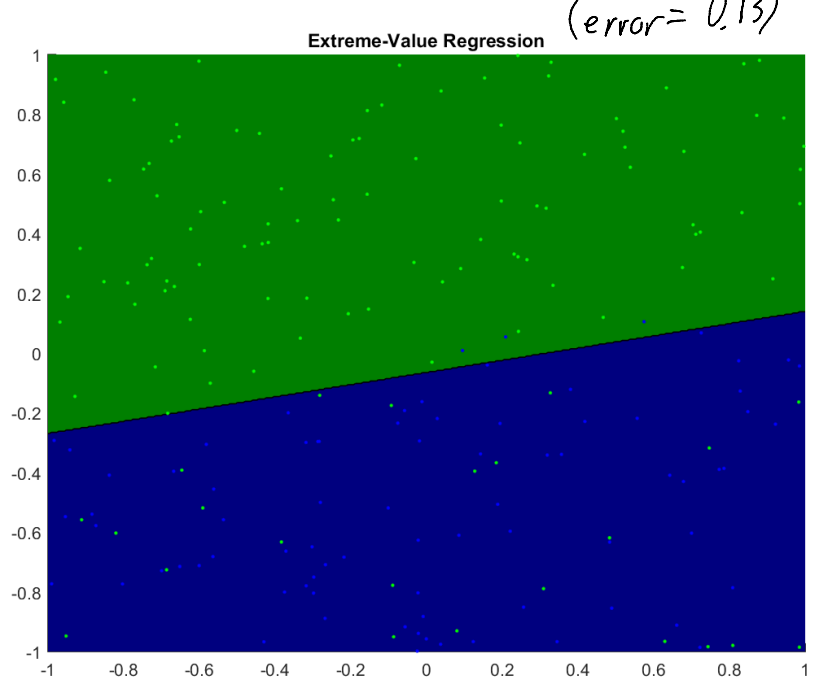
$$p(y_i = +1 \mid \hat{y}_i) = 1 - \exp(-\exp(\hat{y}_i)) \quad [+1 \text{ is majority class}]$$

asymmetric

To make it a probability, $p(y_i = -1 \mid \hat{y}_i) = \exp(-\exp(\hat{y}_i))$

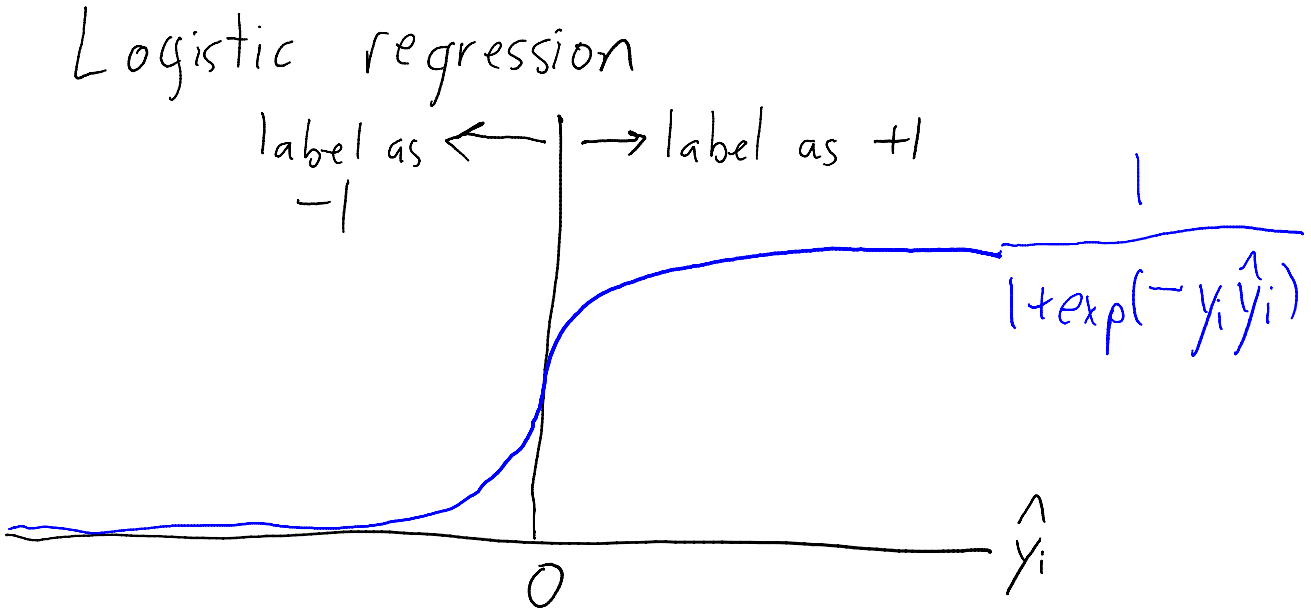


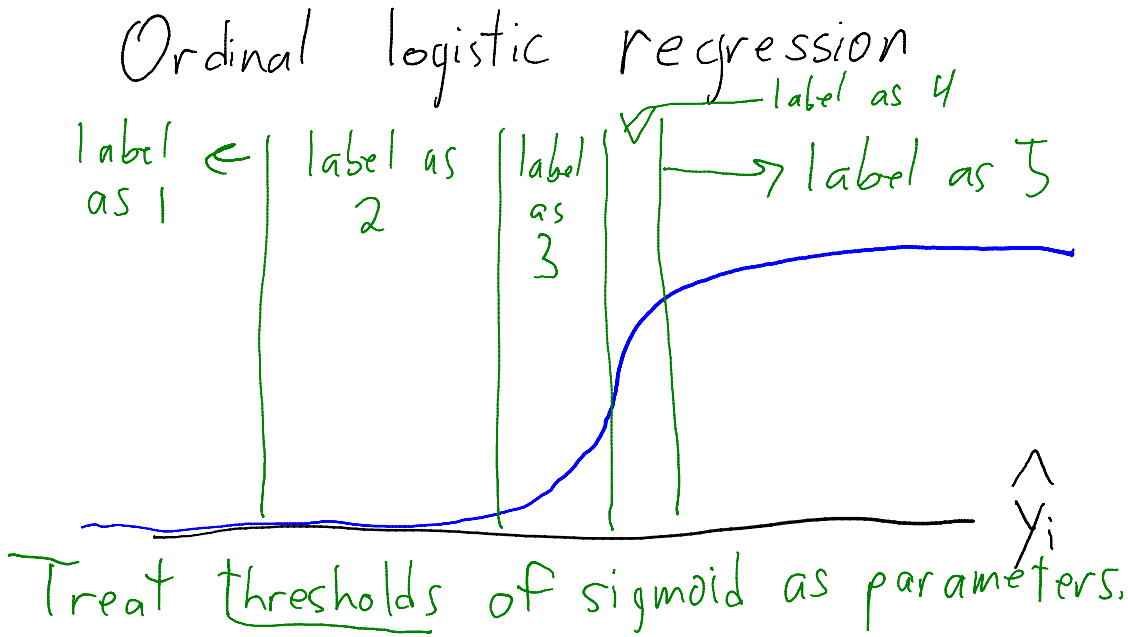




Ordinal Data and Proportional Odds

- Ordinal data: categorical data where the order matters:
  - Rating hotels as {'1 star', '2 stars', '3 stars', '4 stars', '5 stars'}.
- Could do softmax, but softmax ignores the order.
- 'Proportional odds' or 'ordinal logistic regression':





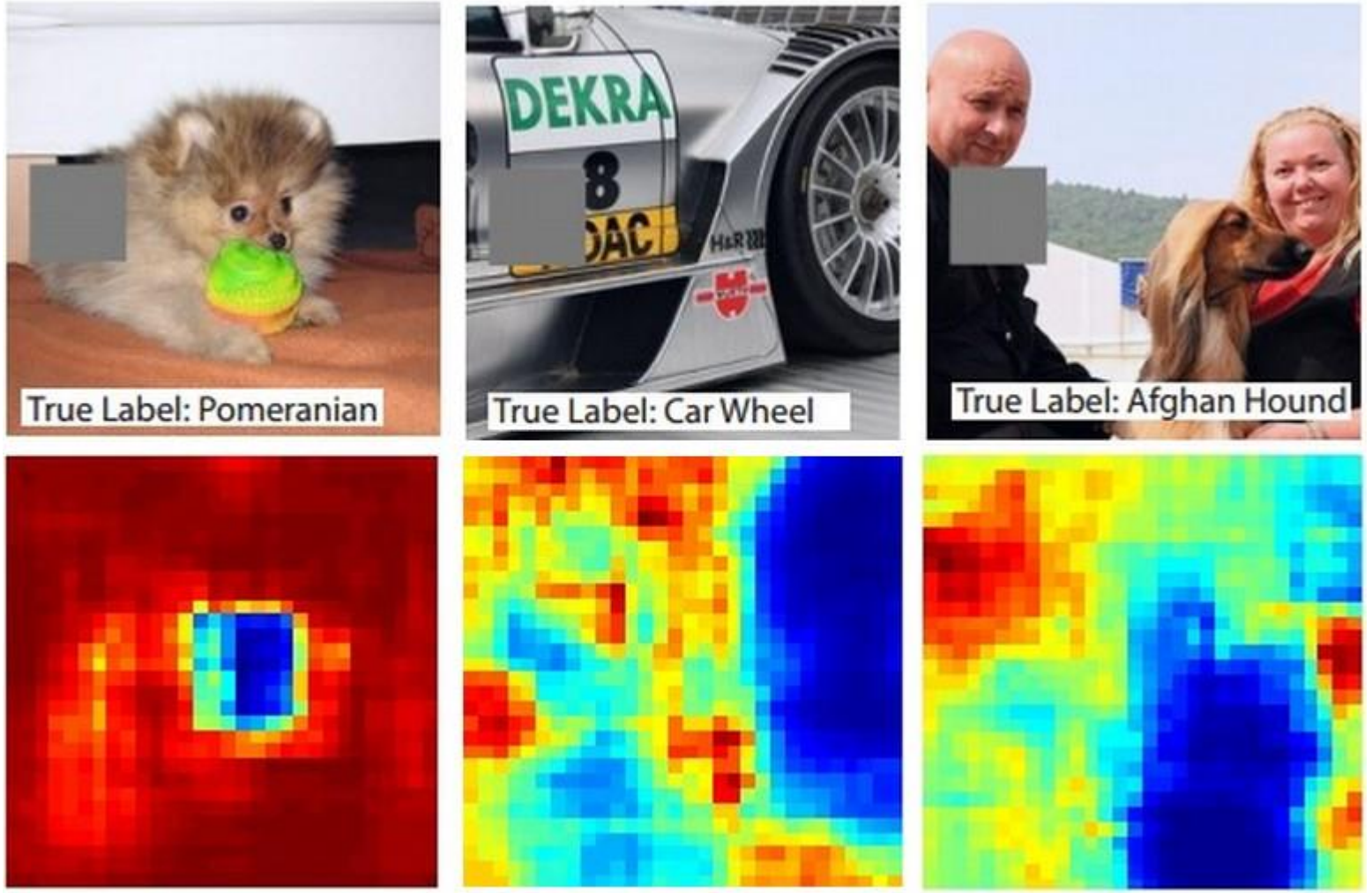
Count Data and Poisson Loss

- Count data: predict the number of times something happens.
  - The number of ‘likes’ that Facebook will get.
- Softmax/ordinal require finite number of categories.
- We probably don’t want separate parameter for ‘654’ and ‘655’.
- Poisson regression: use probability from Poisson count distribution.

Back to discussion of CNNs…

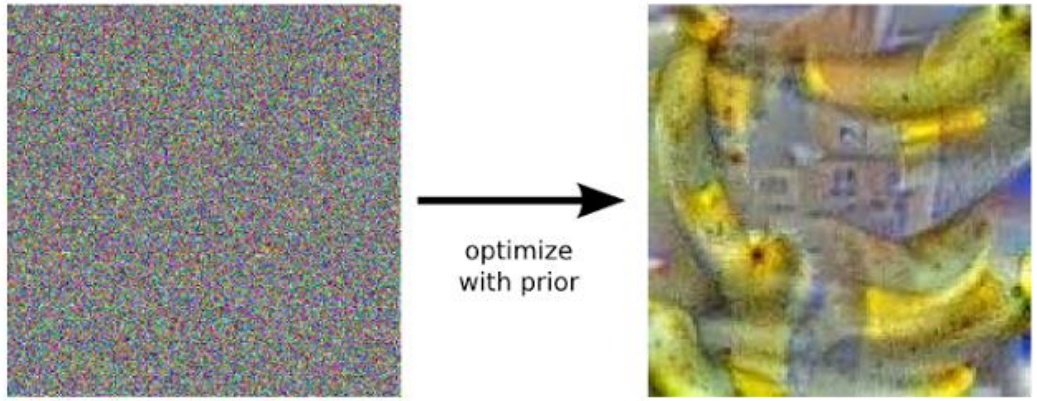
Interpreting CNNs

How does prediction change if we hide part of the image?



Inceptionism

- Start with random noise, move image to amplify class label.
  - (And enforce pairwise statistics on image)



http://googleresearch.blogspot.ca/2015/06/inceptionism-going-deeper-into-neural.html

Inceptionism

- Start with an image, amplify features deep in network.

Starting from random noise:

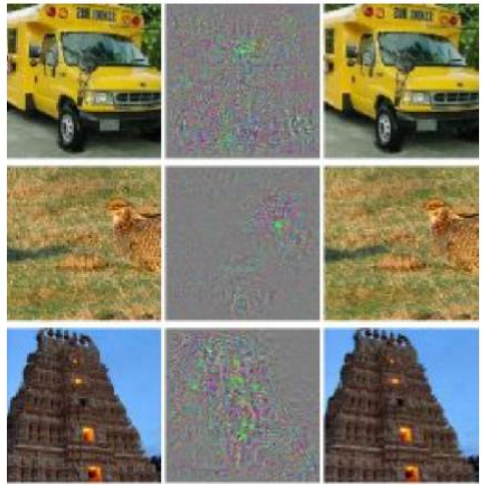
http://googleresearch.blogspot.ca/2015/06/inceptionism-going-deeper-into-neural.html

CNNs for Artistic Style

http://arxiv.org/pdf/1508.06576v2.pdf

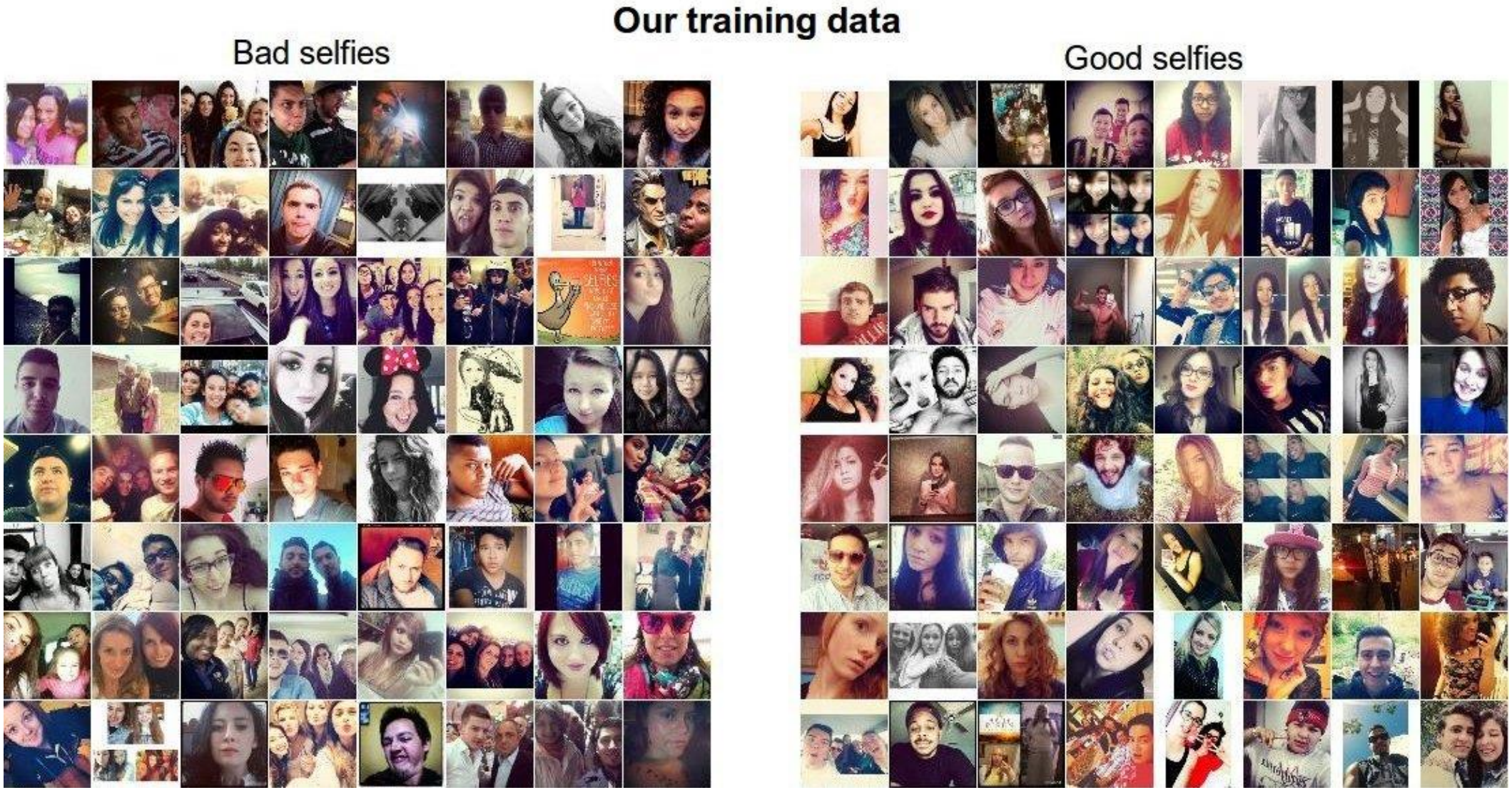
Is this magic?

- For speech recognition and object detection:
  - No other methods have ever given the current level of performance.
  - But, we also don't know how to scale up other universal approximators.
- No baseline methods in many deep learning papers/articles/blogs.
  - Would simpler methods obtain similar performance?
- Despite the 'high-level' abstraction of deep models, easily fooled:
  - But progress on fixing 'blind spots'.



http://cs.nyu.edu/~zaremba/docs/understanding.pdf

CNNs for Rating Selfies

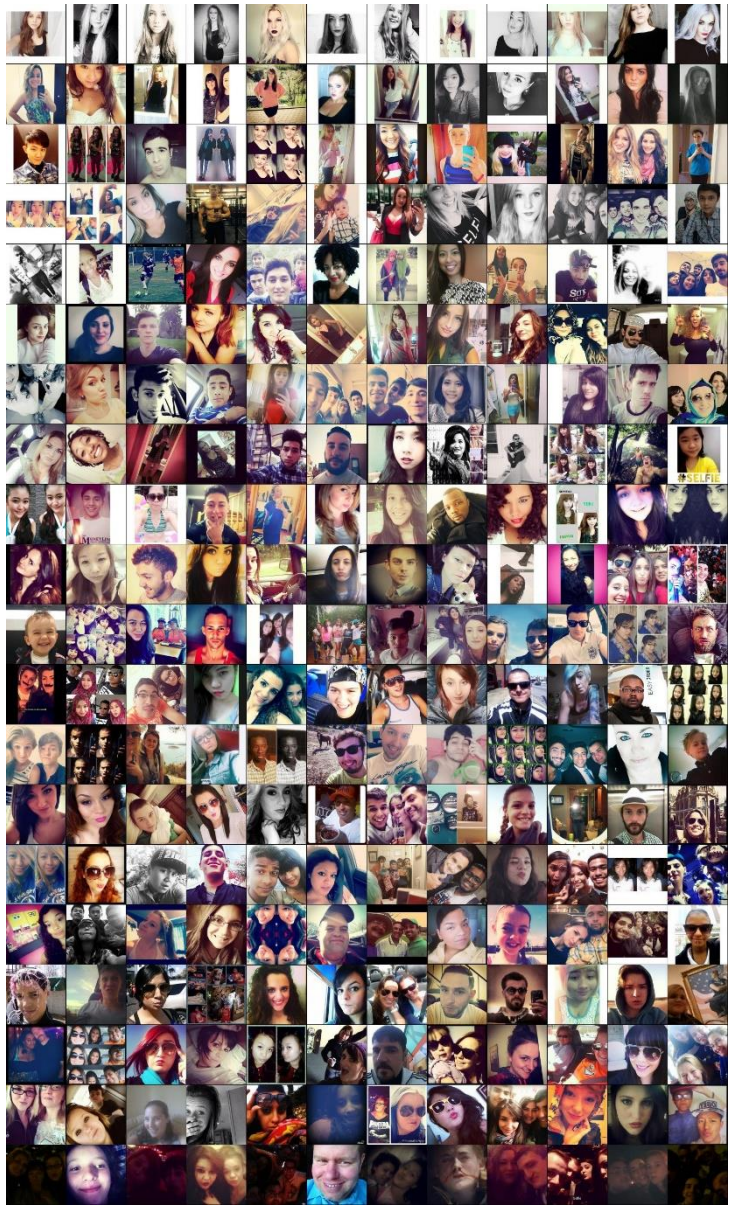


https://karpathy.github.io/2015/10/25/selfie/

CNNs for Rating Selfies

Do:

- Be female
- Face in 1/3 of image
- Cut off forehead
- Show long hair
- Oversaturate face
- Use filter
- Add border



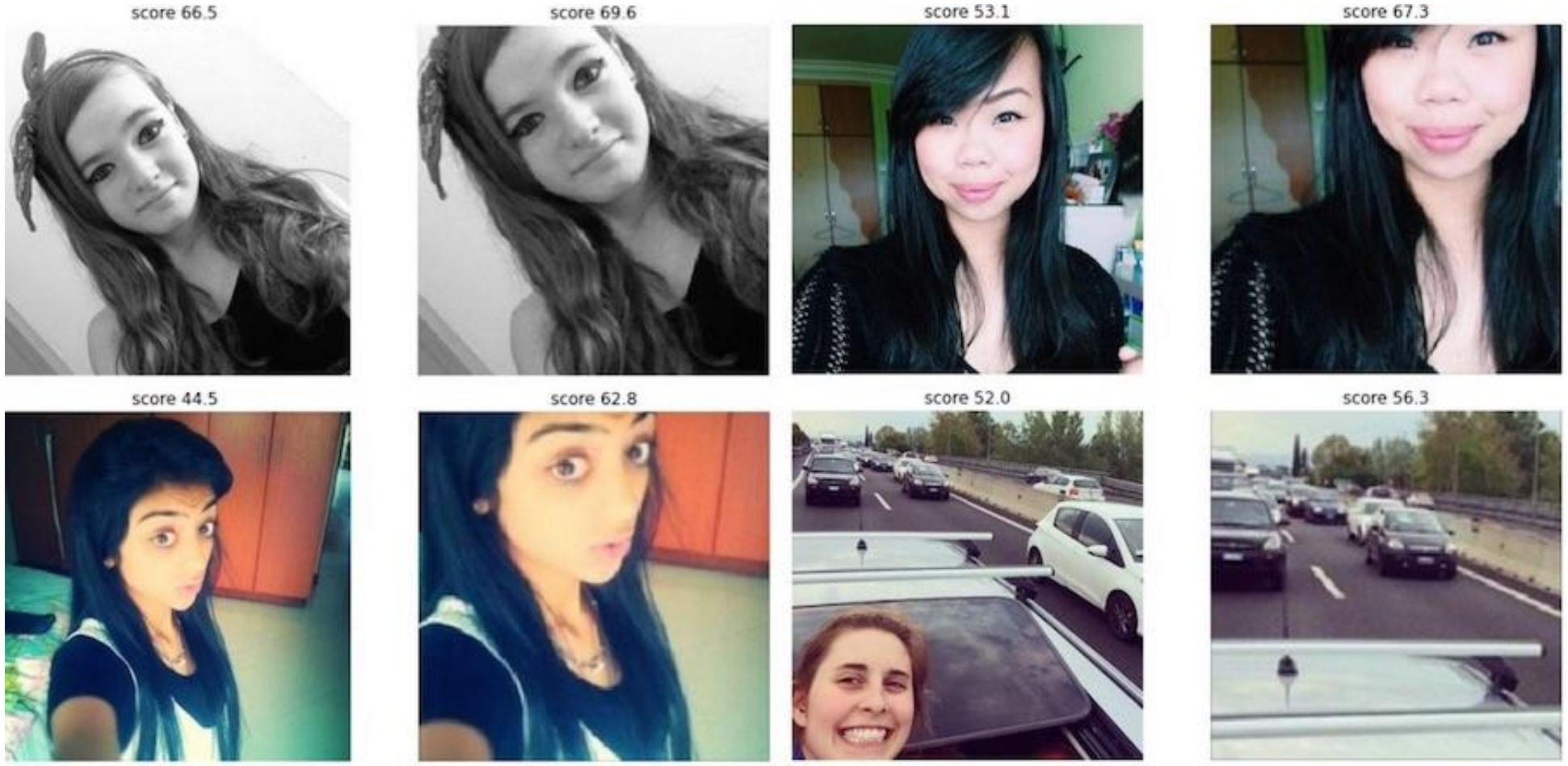
Don't:

- use low lighting
- make head too big
- take group shots

https://karpathy.github.io/2015/10/25/selfie/

CNNs for Rating Selfies

"Best" crop:



https://karpathy.github.io/2015/10/25/selfie/

Summary

- Squared loss is rarely the right measure.
- -log(probability) lets us to define loss from any probability.
- Softmax is natural loss for categorical data.
- Ordinal logistic is natural loss for ordinal data.
- Exotic losses like extreme-value or Poisson for certain situations.

- Next time: finding all the cat videos on YouTube.