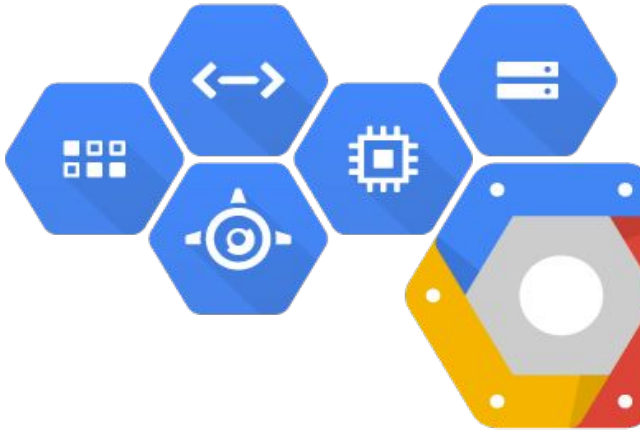
Google Cloud

Google Cloud Analytics 2-Day Training - ML

May 2018



© 2017 Google Inc. All rights reserved.

Objectives

This 2-day training is designed to provide knowledge to deliver GCP's Big Data and Machine Learning capabilities. Attendees will leave the training with the following:

- Knowledge of how to deliver GCP Big Data solutions for our clients
- Deeper understanding of GCP's Big Data services and how they integrate with each other
- Knowledge of how GCP enables the Big Data Reference Architecture
- Understanding of technical design considerations for GCP's data services
- Understanding of how to enable a data pipeline on GCP

It is assumed that the attendees have GCP foundational knowledge for Big Data and Machine Learning

Agenda - Machine Learning

How Google Does Machine Learning	How do we build a foundation for Machine Learning discovery?	**8:00 - 9:30 AM**
	How do we identify and qualify potential Machine Learning applications?	
	What are the phases of machine learning models	
Art & Science of Machine Learning	What are the factors that influence design decisions for develop machine learning	**9:30 - 10:30 AM**
Break		**10:30 - 10:40 AM**
GCP Tools for ML	What tools can we use to develop Machine Learning application with GCP?	**10:40 - 12:00 PM**
	What are value add capabilities built into each of the tools?	
	In what cases do we use each tool set for Machine Learning?	
Lunch		**12:00 - 1:00 PM**
Deep Dive: Developing an End to End ML Solution	Demo: Predictive maintenance with Machine Learning?	**1:00 - 3:00 PM**
	How does a team develop an end to end Machine Learning solution with GCP?	
Recap	Summary of key topics	

Section 1 Section 2 Section 3 Section 4

How Google Does Machine Learning

Derived from talks by Ian Goodfellow and Martin Zinkevich

Proprietary + Confidential

Themes

1. Keep it simple!
2. Do ML like the great engineer you are, not the ML expert you aren't.

© 2017 Google Inc. All rights reserved.

Format

- These slides are derived from talks by Ian Goodfellow (Practical Methodology for Deploying Machine Learning) and Martin Zinkevich (Effective Machine Learning)
- In this talk, we'll outline a step by step methodology for building a machine learning system.
- For each step, we'll discuss common pitfalls, myths, and rules. All 44 rules are available here: https://developers.google.com/machine-learning/rules-of-ml/.

© 2017 Google Inc. All rights reserved.

How Google Does Machine Learning

1. Use needs to define metric-based goals
2. Build an end-to-end system
3. Data-driven refinement

© 2017 Google Inc. All rights reserved.

Step #1: Use needs to define metric-based goals

Proprietary + Confidential

Understand Your Requirements

- Does your system need to have high accuracy? Or is it OK to be wrong sometimes?
  - Surgery robot. Not valuable unless near-perfect accuracy.
  - Recommendation system. Still valuable at low accuracy levels.

- Common metrics
  - Accuracy
  - Precision
  - Recall
  - Coverage

- Once you understand your requirements, *build the simplest system possible to meet those requirements.*

© 2017 Google Inc. All rights reserved.

Rule #1: Don’t be afraid to launch a product without ML

© 2017 Google Inc. All rights reserved.

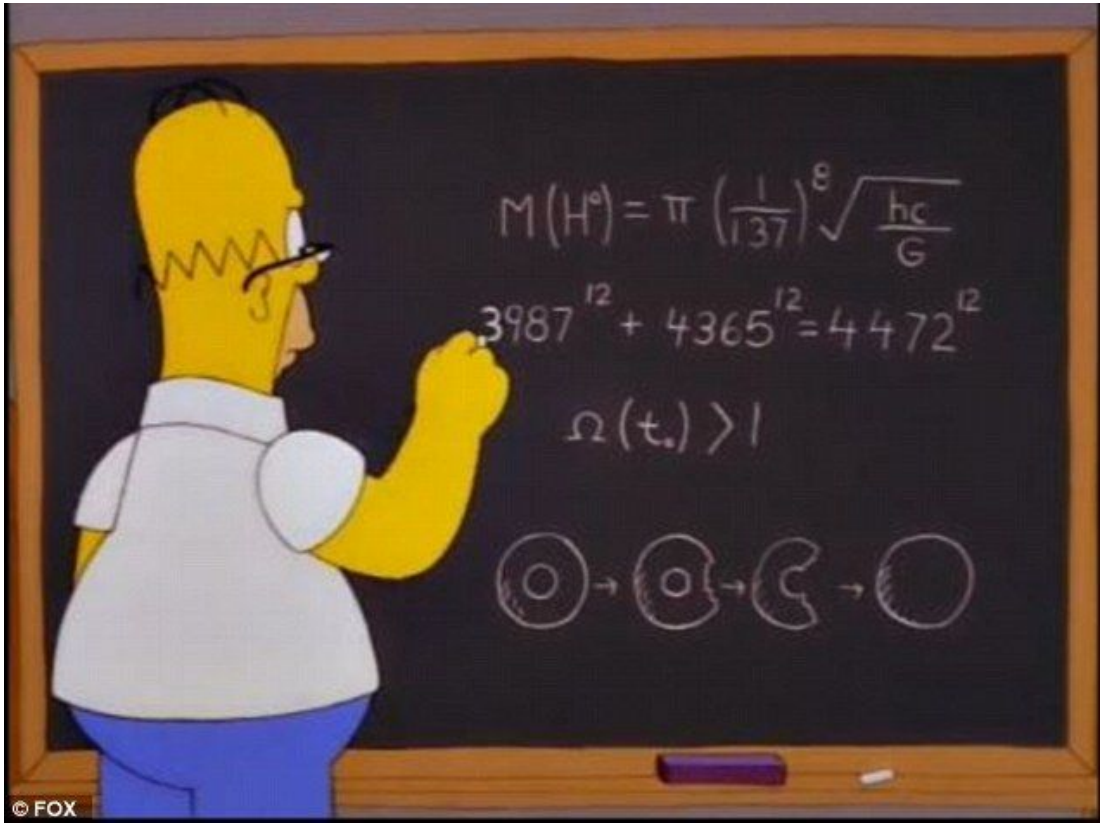
Google Cloud

© 2017 Google Inc. All rights reserved.

Rule #3: Choose ML over a complex heuristic.

© 2017 Google Inc. All rights reserved.

If your heuristic looks like this, it’s time to move to ML.



© 2017 Google Inc. All rights reserved.

Step #2: Build end-to-end system

Proprietary + Confidential

Make sure your data doesn’t suck

- The data you use to train your model is the single most important factor in determining the success of your ML project.
- I often spend > 50% of my time on an ML project just looking at data.
  - Visualize the data
  - Use common sense
- Common pitfalls: poorly documented fields, incorrect or missing values, lack of labels, missing important signals, some classes don’t have enough examples.
- **If your data sucks, fix it before building an ML system.**
  - Would you build a rocket if you knew you didn’t have any rocket fuel?

© 2017 Google Inc. All rights reserved.

Rule #4: Keep the first model simple and get the infrastructure right

© 2017 Google Inc. All rights reserved.

Use the highest level abstraction possible

Proprietary + Confidential

Getting the infrastructure right

...by being the great engineer you are, not the ML expert you aren't

- Choose a few, simple features to start
- Make sure features reach your learning algorithm correctly
- Write good tests (Rule #5: Test the infrastructure independently from the machine learning)
- Encode your data validation logic in the pipeline
- Monitoring
  - Rule #8: Know the freshness requirements of your system
  - Rule #9: Detect problems before exporting models
  - Rule #11: Give feature sets owners and documentation (for real, document your features)
- General software engineering best practices: inversion of control, single responsibility, separation of concerns, continuous integration, etc

© 2017 Google Inc. All rights reserved.

Myth #1: You have to know a bunch of fancy algorithms to be a successful ML practitioner

© 2017 Google Inc. All rights reserved.

Creating your first model

...and keeping it simple

Rule #12: Don’t overthink which objective you choose to directly optimize

Rule #13: Choose a simple, observable, and attributable metric for your first objective

Rule #14: Starting with an interpretable model (*cough* Linear/Logistic Regression) makes debugging easier

© 2017 Google Inc. All rights reserved.

Step #3: Data driven refinement

Proprietary + Confidential

Data driven refinement

Once you have an end to end system, and have an established baseline, all that's left to do is improve performance iteratively until you've reached the metrics you decided on in Step #1.

As complicated as machine learning is, there are really only two options available:

1. Create new features

2. Add complexity to your model

Make sure you can run experiments quickly. Data driven refinement won't be fun if each iteration takes a day.

© 2017 Google Inc. All rights reserved.

Feature Engineering

Above all, use common sense when creating features.

Rule #20: Combine and modify features to create new features in human-understandable ways.

Rule #21: The number of feature weights you can have in a model is roughly proportional to the amount of data you have.

Rule #22: Clean up features when you are no longer using them (Corollary: Add features in a common sense way. Don’t just throw a zillion features at your model and hope for the best)

© 2017 Google Inc. All rights reserved.

Modeling Approach

Linear models can only take you so far. Once you've got your infrastructure figured out (and if your data has some nonlinear patterns), try something a little more powerful.

- Support Vector Machine
- Boosted Tree / Random Forest
- Neural Network (gasp)

As long as you have enough data, adding complexity to your model (for example, adding more layers to a neural network), can lead to significant improvements in model performance. Just beware of overfitting.

© 2017 Google Inc. All rights reserved.

ML Fairness

Machine learning models learn only on the data you provide. It is essential you provide your model an unbiased dataset to learn from in order to produce a fair model.

- Be careful when including features like age, gender, or race in your training data. This is sometimes appropriate (for example, when training a model to predict a certain disease) but should generally be avoided.
- It's usually not so obvious. For example, including credit score in a model can have undesirable side effects.
- Use frameworks like TensorFlow Model Analysis to make sure your model is working for all demographics, not just the majority.

© 2017 Google Inc. All rights reserved.

Additional resources

1. What's your ML Test Score? https://ai.google/research/pubs/pub45742
2. Effective Machine Learning. http://martin.zinkevich.org/rules_of_ml/rules_of_ml.pdf
3. Hidden Technical Debt in Machine Learning Systems. https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf

© 2017 Google Inc. All rights reserved.

Art & Science of Machine Learning

Proprietary + Confidential

Art & Science of Machine Learning

Discussion in this section involves the following topics

1. Fundamental Concepts of ML
2. A Structured Approach to Machine Learning

© 2017 Google Inc. All rights reserved.

Fundamental Concepts of ML

© 2017 Google Inc. All rights reserved.

Agenda

Supervised and Unsupervised Machine Learning

Regression and Classification

Loss and Gradient Descent

Hyperparameters (learning rate, batch size)

Accuracy Metrics

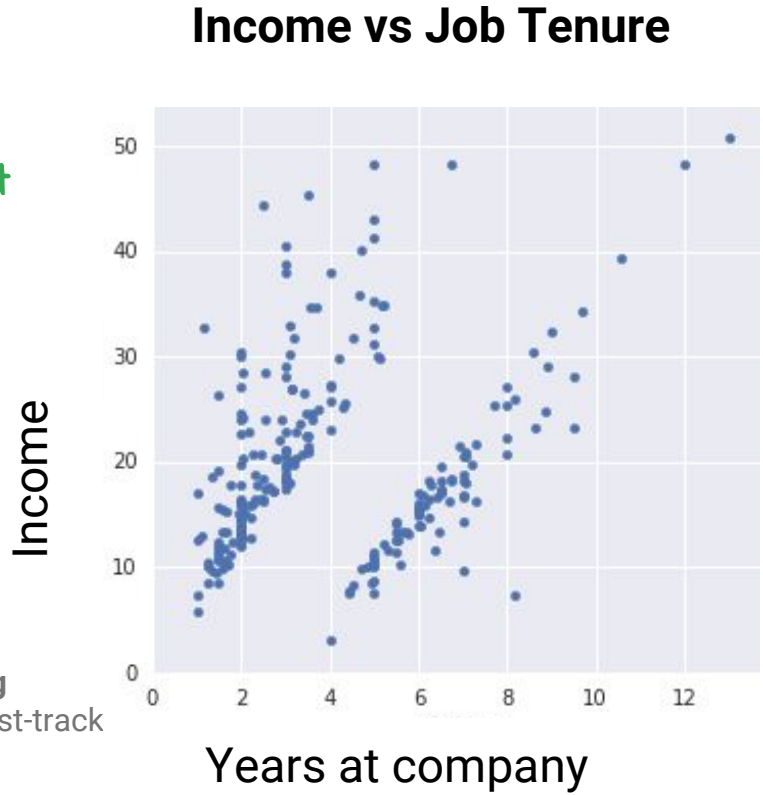
Preventing Overfitting

Intro to Neural Nets

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Unsupervised and supervised learning are the two types of ML algorithms

In unsupervised learning, data is not labeled

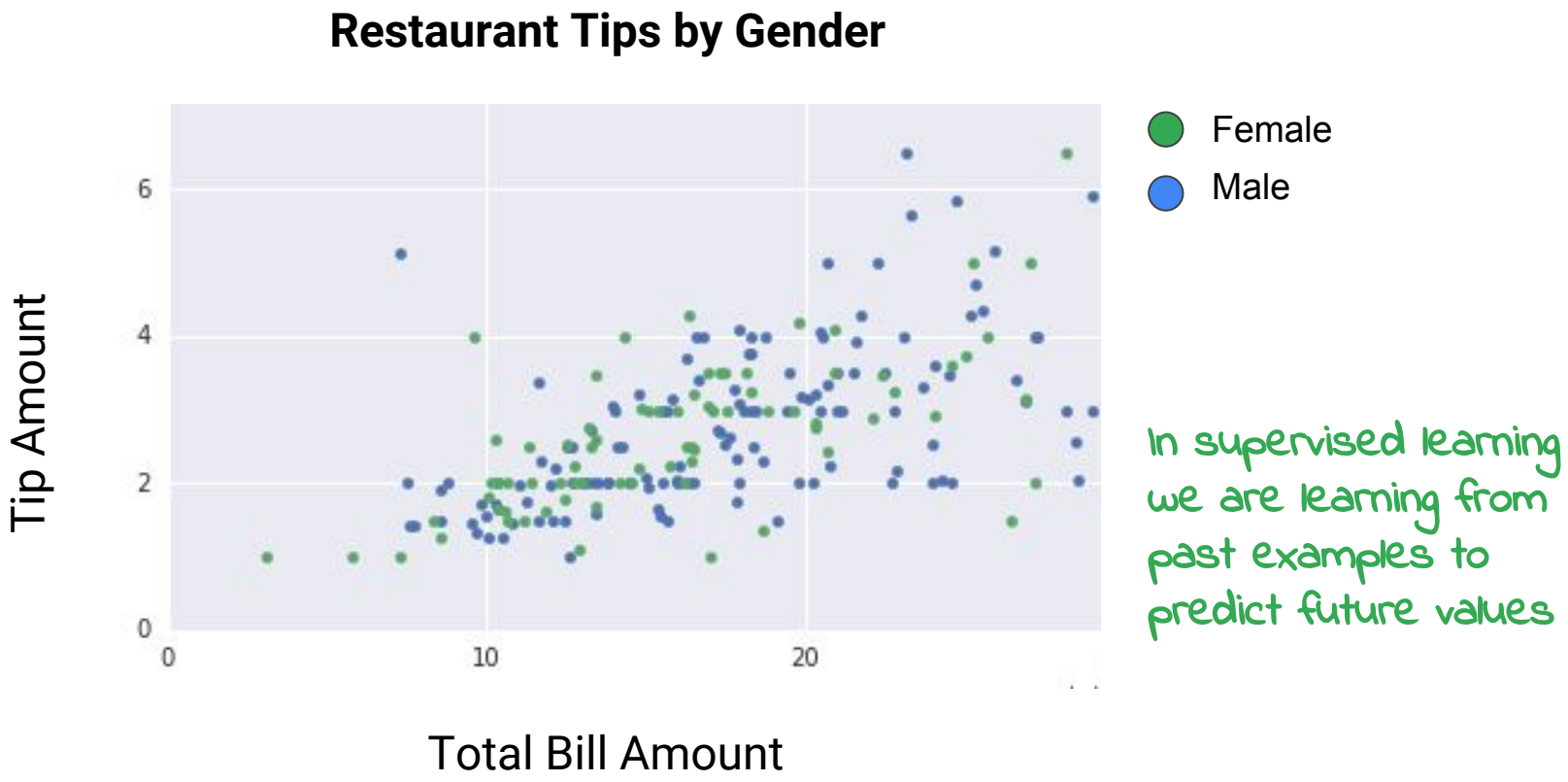


Example Model: Clustering
Is this employee on the "fast-track or not?

31

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

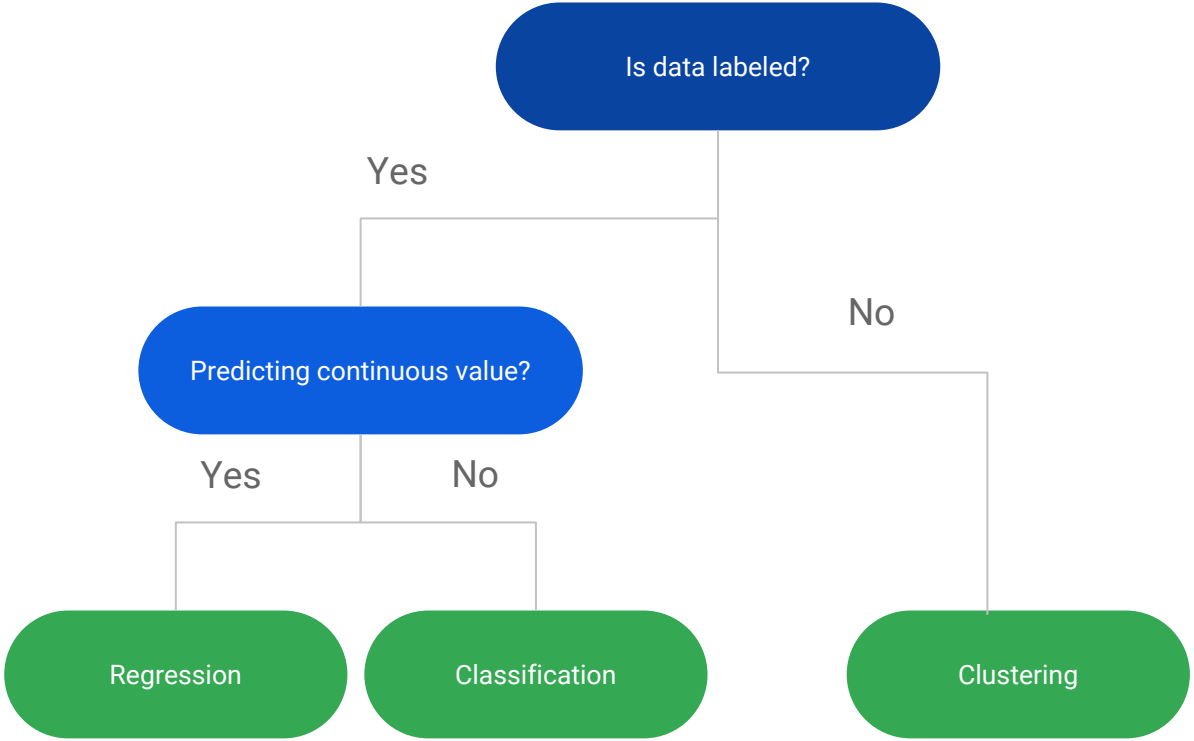
Supervised learning implies the data is already labeled



In supervised learning we are learning from past examples to predict future values

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

The type of ML problem depends on whether or not you have labeled data and what you are interested in predicting



© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Agenda

- Supervised and Unsupervised Machine Learning
- Regression and Classification
- Loss and Gradient Descent
- Hyperparameters (learning rate, batch size)
- Accuracy Metrics
- Preventing Overfitting
- Intro to Neural Nets

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Regression and classification are supervised ML model types

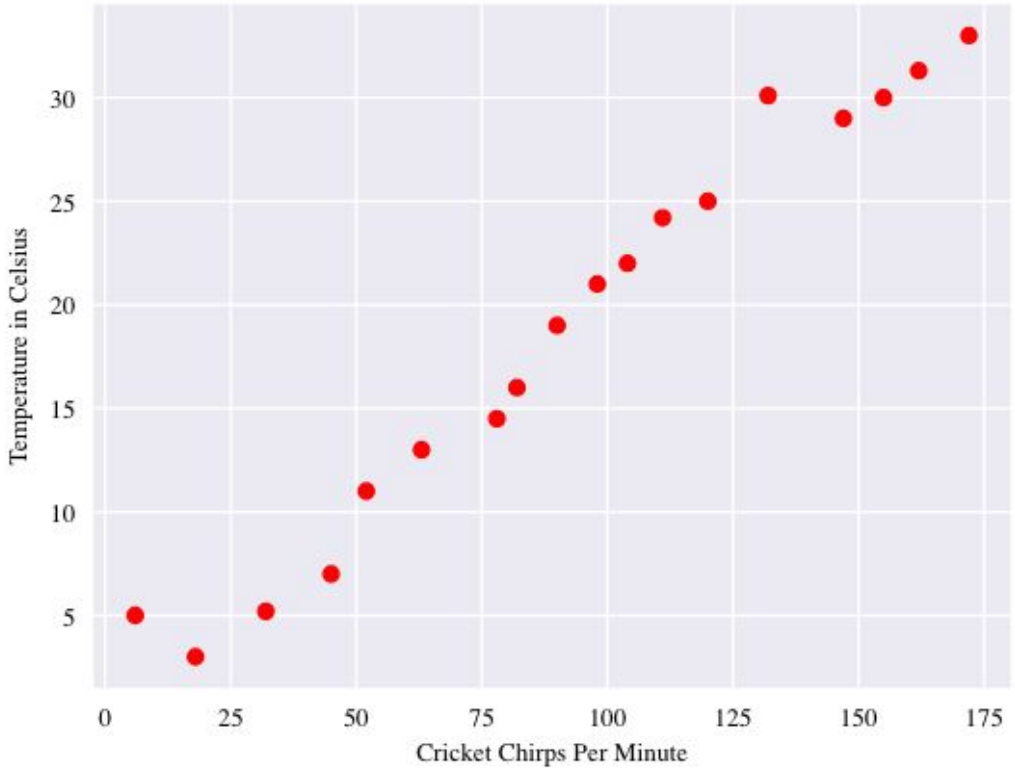
	total_bill	tip	sex	smoker	day	time
2	16.99	1.01	Female	No	Sun	Dinner
3	10.34	1.66	Male	No	Sun	Dinner
4	21.01	3.5	Male	No	Sun	Dinner
5	23.68	3.31	Male	No	Sun	Dinner
6	24.59	3.61	Female	No	Sun	Dinner
7	25.29	4.71	Male	No	Sun	Dinner
8	8.77	2	Male	No	Sun	Dinner
9	26.88	3.12	Male	No	Sun	Dinner

Option 1
Regression Model
Predict the tip amount

Option 2
Classification Model
Predict the sex of the customer

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Linear Regression



© 2017 Google Inc. All rights reserved.

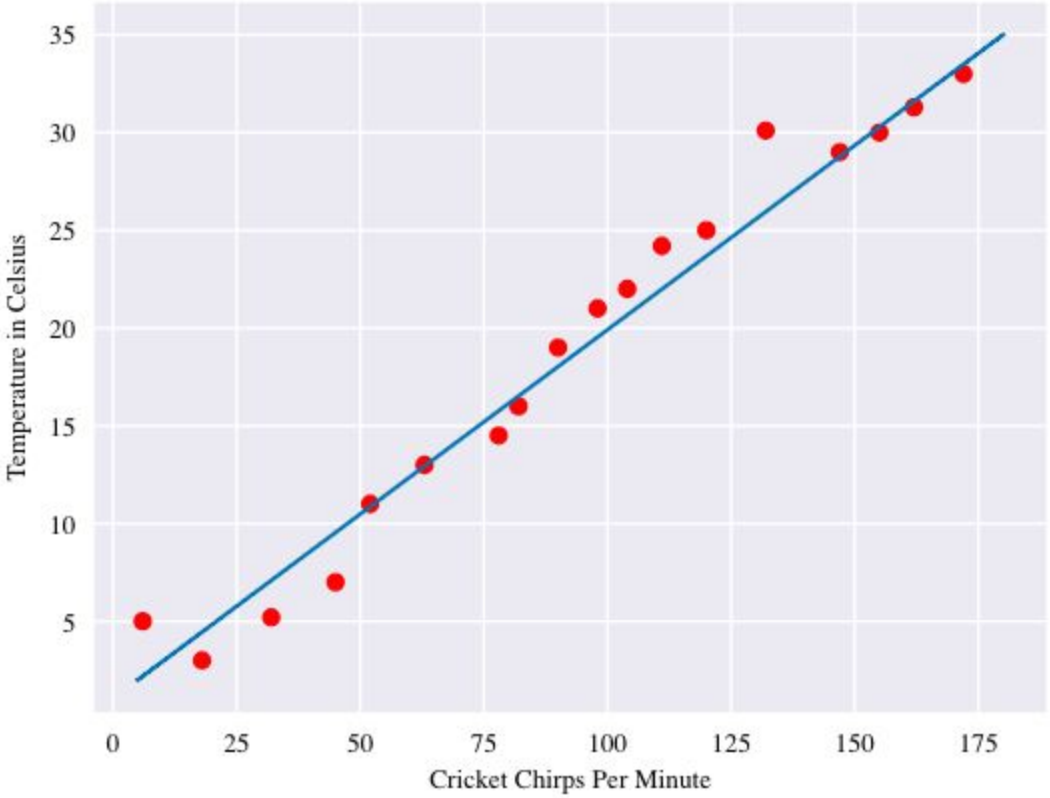
Linear Regression

Equation for points:

$$y = mx + b$$

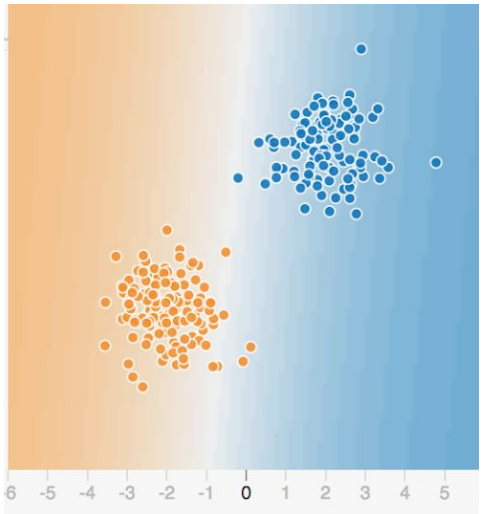
...In the context of ML:

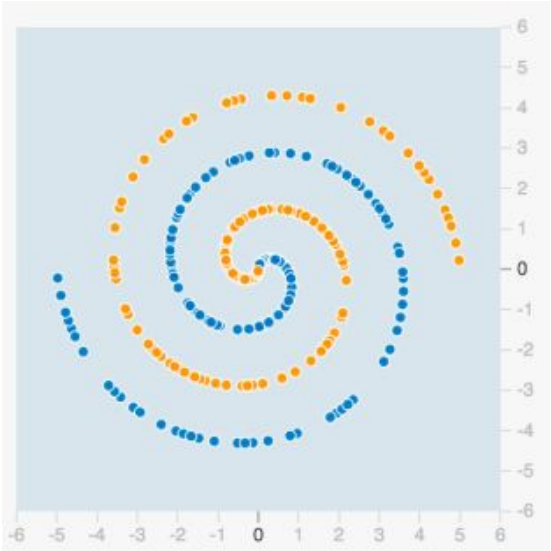
$$y' = b + w_1 x_1$$



© 2017 Google Inc. All rights reserved.

Classification Visual Intuition





Class 1
Class 2

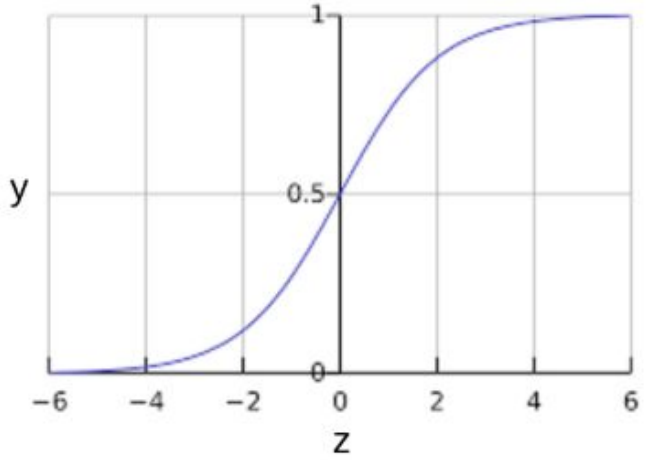
Google Cloud

© 2017 Google Inc. All rights reserved.

Classification via Logistic Regression

Sigmoid Function

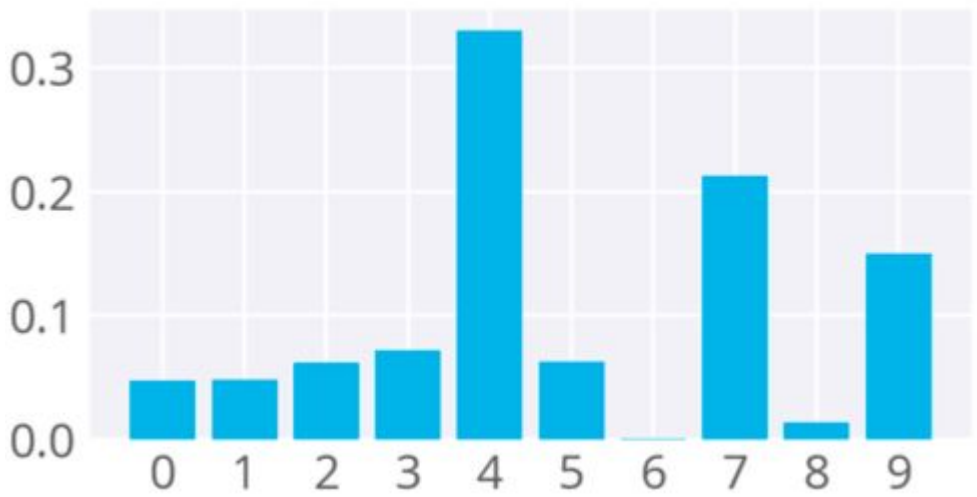
$$z = log(\frac{y}{1-y})$$



continuous prediction value

i.e. $\hat{y} = w_0x_0 + w_1x_1 + w_2x_2 + ...$

© 2017 Google Inc. All rights reserved.



Label	0	1	2	3	4	5	6	7	8	9
Array	[0,	0,	0,	0,	1,	0,	0,	0,	0,	0]

Probabilities for each class

in the form of a one-hot encoded vector

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Quick Quiz!

Imagine you are in banking and you are creating an ML model for detecting if transactions are fraudulent or not. Is this classification or regression and why?

A. Regression, categorical label
B. Regression, continuous label
C. Classification, categorical label
D. Classification, continuous label

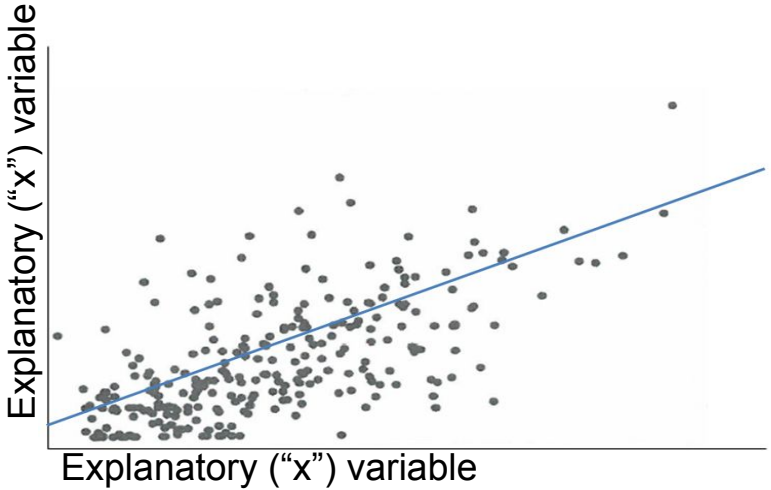
© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

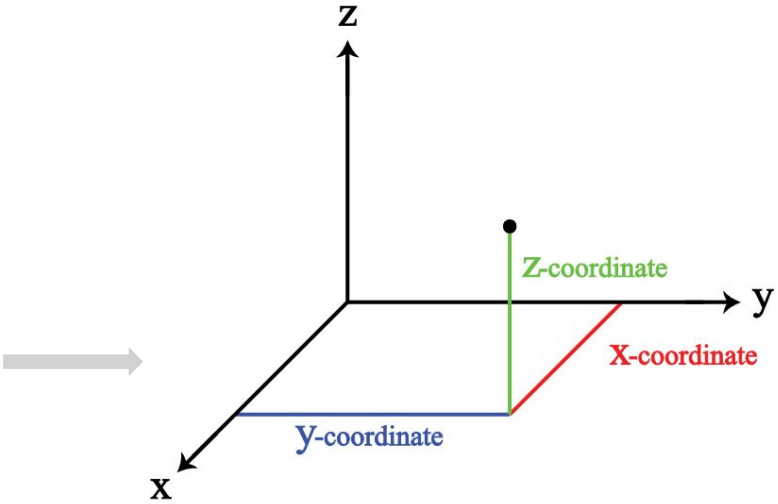
Agenda

- Supervised and Unsupervised Machine Learning
- Regression and Classification
- Loss and Gradient Descent
- Hyperparameters (learning rate, batch size)
- Accuracy Metrics
- Preventing Overfitting
- Intro to Neural Nets

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

How Multiple Dimensions are Represented





Single variable:

$$y = mx + b$$

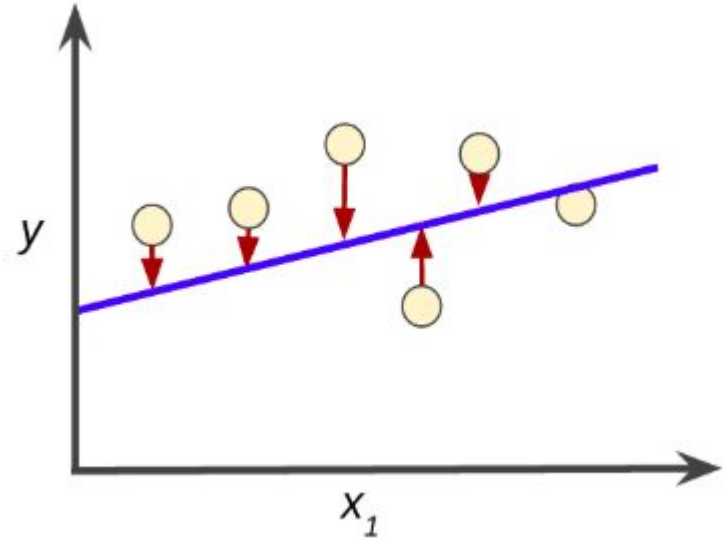
$$y = \beta_1 x_1 + \alpha$$

Multiple variables:

$$y = \beta_1 x_1 + \beta_2 x_2 + \ldots + \alpha$$

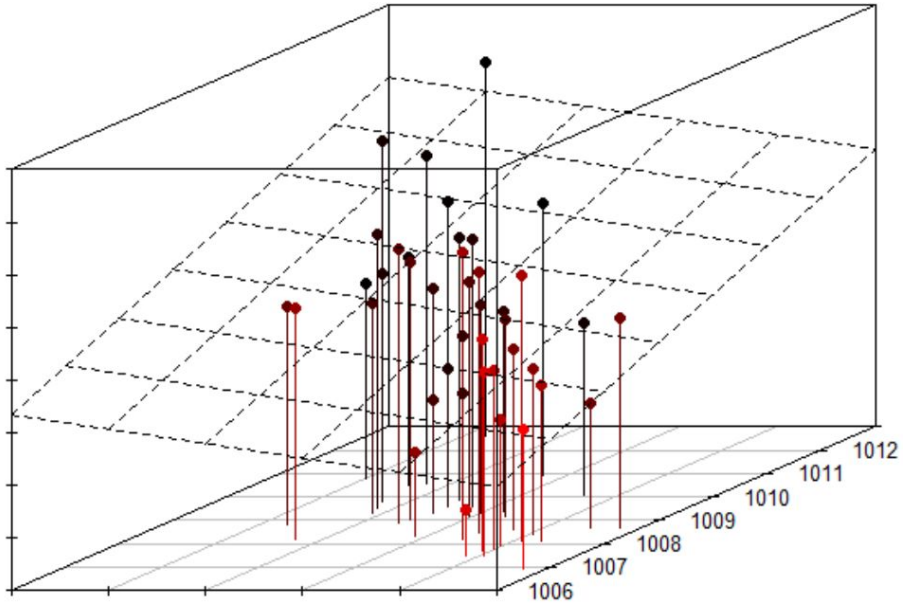
© 2017 Google Inc. All rights reserved.

It’s Not Perfect!



© 2017 Google Inc. All rights reserved.

Error in multiple dimensions



© 2017 Google Inc. All rights reserved.

In linear regression, the prediction is a weighted sum of inputs

$$\hat{y} = w_0 x_0 + w_1 x_1 + w_2 x_2 + \ldots$$

$$\hat{y} = X\omega$$

This can be condensed down into a simple matrix equation

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Equation for a linear model tying mother's age and baby weight

The slope of the line is given by w_1

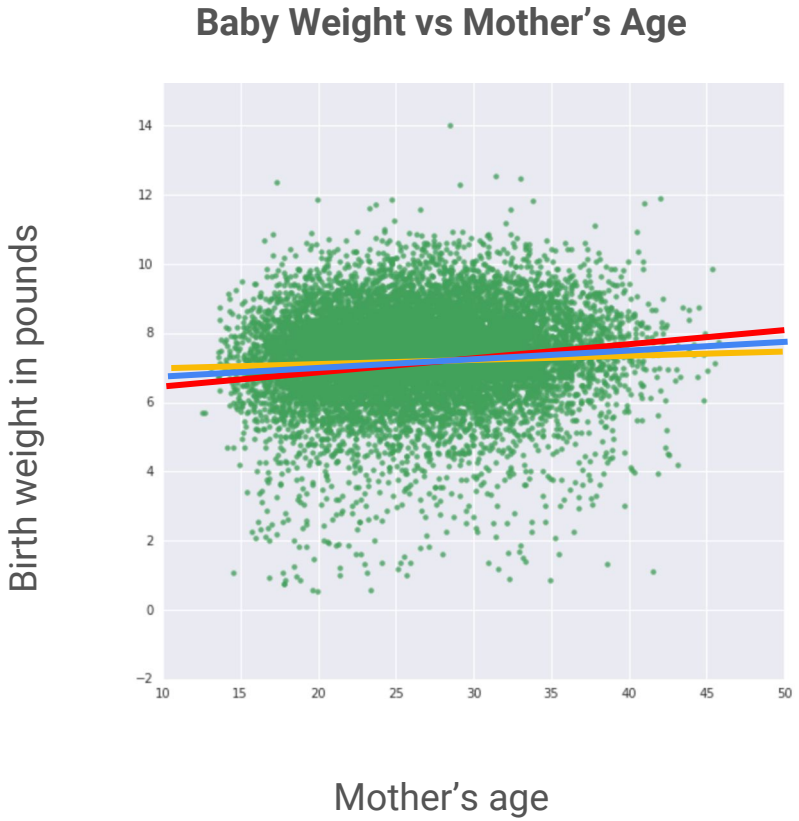
$$y = w_1 x_1 + b$$

- x_1 is the **feature** (e.g. mother's age)
- w_1 is the **weight** for x_1

Line: y= .02x + 6.83

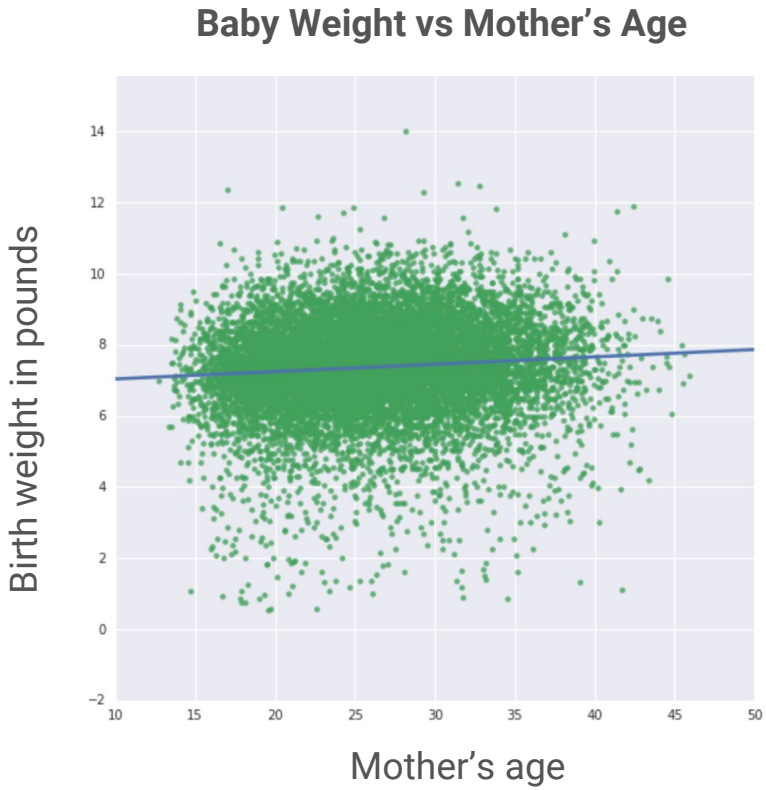
Line: y= .03x + 6.49

Line: y= .01x + 7.14



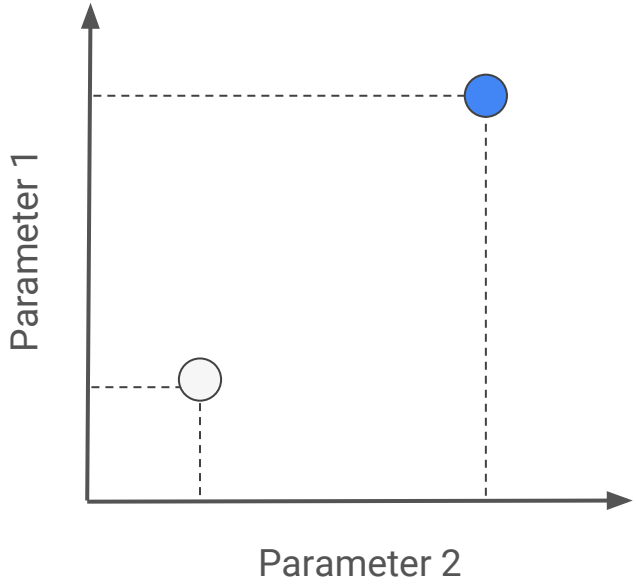
© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Can’t we just solve the equation using all the data?



© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Searching in Parameter-Space



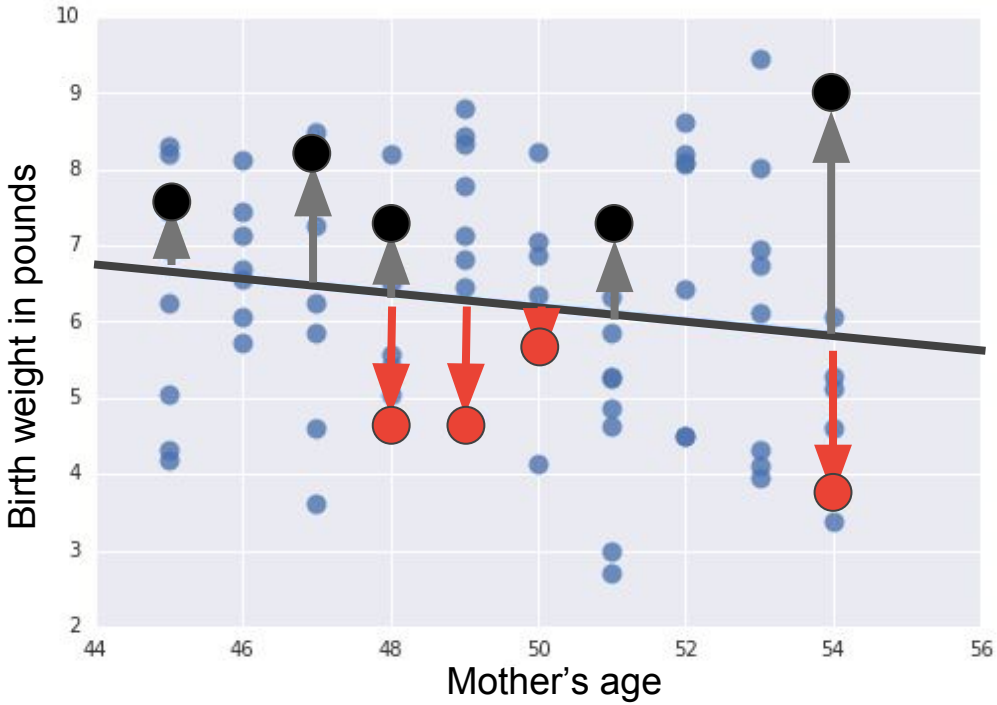
© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Composing a loss function by calculating errors

Error = actual (true) - predicted value
Compute the errors:

```
+0.70
+1.10
+0.65
-1.20
-1.15
+1.10
+3.09
-2.10
```

Each error makes sense. How about all the errors added together?



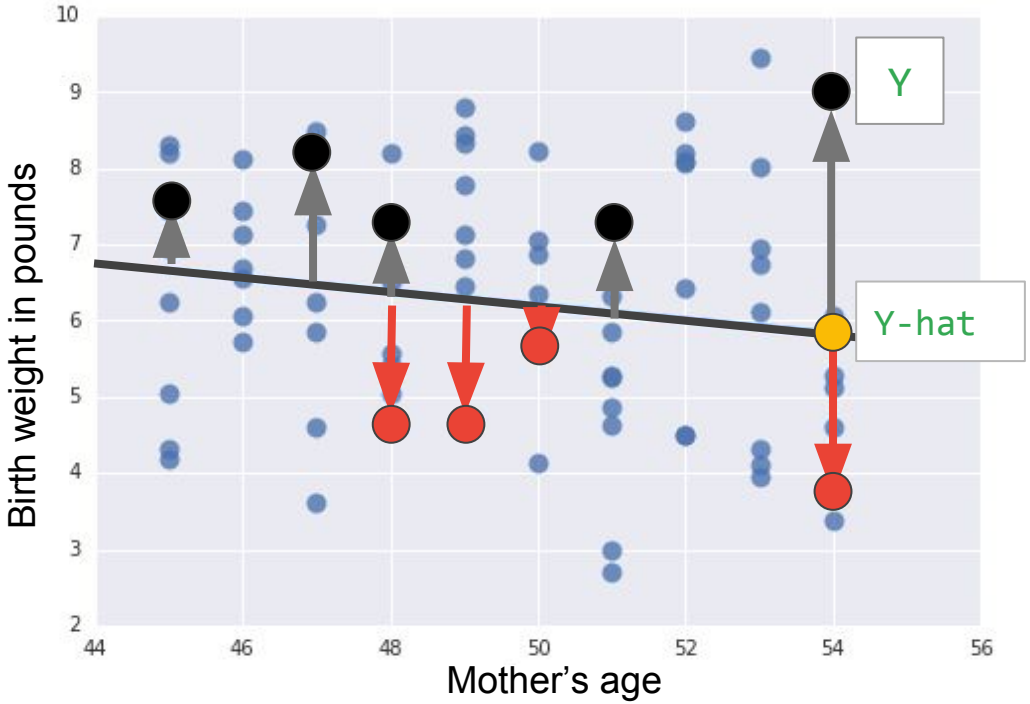
© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Review: (R)MSE is a good loss function for linear regression

$$\sqrt{\frac{1}{n} \times \sum_{i=1}^{n} (\hat{Y}_i - Y_i)^2}$$

Y-hat is the model's estimate

Y is the labeled value



© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

One loss function metric is Root Mean Squared Error (RMSE)

1. Get the errors for the training examples

+0.70
+1.10
+0.65
-1.20
-1.15
+1.10
+3.09
-2.10

2. Compute the squares of the error values

0.49
1.21
0.42
1.44
1.32
1.21
9.55
4.41

3. Compute the mean of the squared error values

2.51

4. Take a square root of the mean

1.58

$$\sqrt{\frac{1}{n} \times \sum_{i=1}^{n} (\hat{Y}_i - Y_i)^2}$$

$\hat{Y}_i$ predicted value

Y_i labeled value

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

The typical loss for linear regression is mean squared error

$$L = \|y - X\omega\|^2$$

This is the L2 norm of the residuals squared.

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Loss Metrics

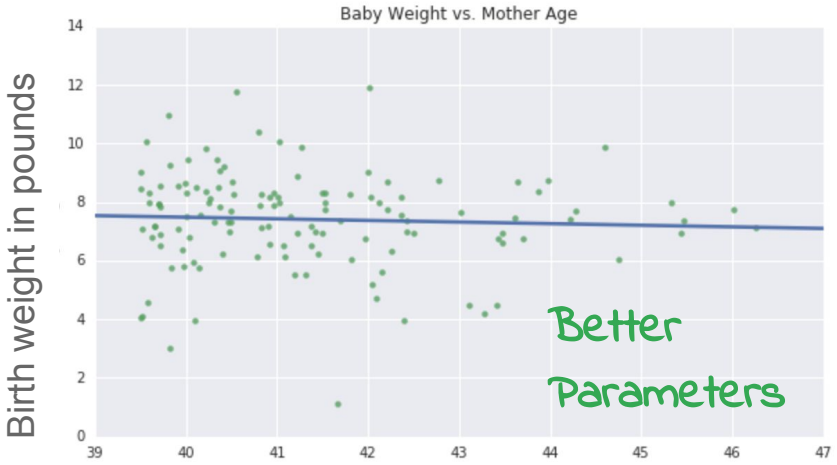
Mean square error (MSE) is the average squared loss per example over the whole dataset. To calculate MSE, sum up all the squared losses for individual examples and then divide by the number of examples:

```
= the square of the difference between the label and the prediction
= (observation - prediction(x))^2
= (y - y')^2
```

$$MSE = \frac{1}{N} \sum_{(x,y) \in D} (y - prediction(x))^2$$

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Lower RMSE indicates a better performing model



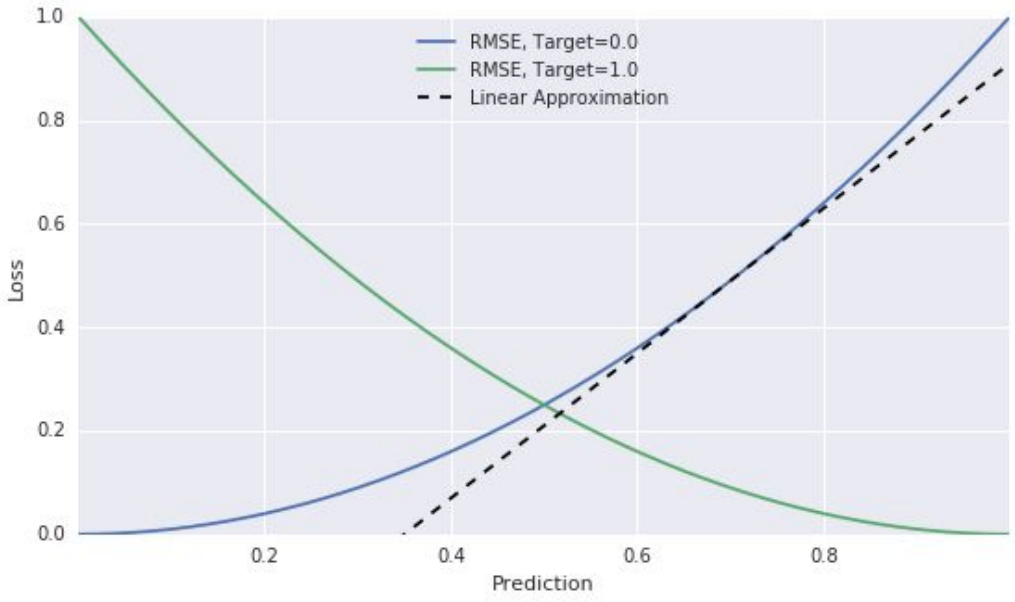
RMSE=.145

RMSE=.149

Need a way to find the best values for weight and bias

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Problem: RMSE doesn't work as well for classification



RMSE doesn't penalize bad classifications appropriately.

56

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Cross Entropy Loss (Softmax) For Classification

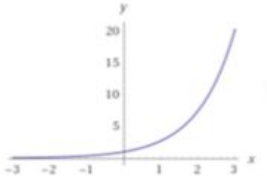
$$s = f(x_i, W)$$

$$P(Y = k|X = x_i) = e^{s_{y_i}} / \sum_j e^{s_j}$$

Each output from the regression equation will be a continuous number from -inf to inf

	Scoring Function	Unnormalized Probabilities
Dog	-3.44	0.03
Cat	1.16	3.19
Boat	-0.81	0.44
Airplane	3.91	49.90

1st step: find the exponential so that number becomes positive



...This creates our unnormalized probabilities

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Cross Entropy Loss (Softmax) For Classification

$$P(Y = k|X = x_i) = e^{s_{y_i}} / \sum_j e^{s_j}$$

	Scoring Function	Unnormalized Probabilities	Normalized Probabilities
Dog	-3.44	0.0321	0.0006
Cat	1.16	3.1899	0.0596
Boat	-0.81	0.4449	0.0083
Airplane	3.91	49.8990	0.9315

For each probability, divide by the sum of the unnormalized probabilities

...This gives normalized probabilities, between 0 and 1

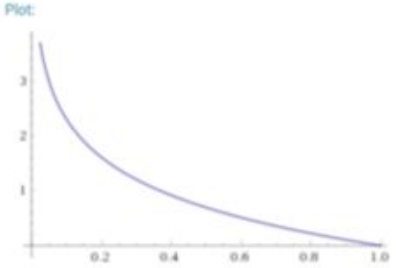
© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Cross Entropy Loss (Softmax) For Classification

$$L_i = -log(e^{s_{y_i}} / \sum_j e^{s_j})$$

	Scoring Function	Unnormalized Probabilities	Normalized Probabilities	Negative Log Loss
Dog	-3.44	0.0321	0.0006	
Cat	1.16	3.1899	0.0596	
Boat	-0.81	0.4449	0.0083	
Airplane	3.91	49.8990	0.9315	0.070

Take the negative log of the normalized probability of the correct class.



...This gives us the final loss

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Cross Entropy Loss (Softmax)For Classification

$$s = f(x_i; W)$$

cat	**3.2**
car	5.1
frog	-1.7

unnormalized log probabilities

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Cross Entropy Loss (Softmax)For Classification

$$L_i = -\log\left(\frac{e^{sy_i}}{\sum_j e^{sj}}\right)$$

unnormalized probabilities

cat	**3.2**	exp	**24.5**
car	5.1		164.0
frog	-1.7		0.18

unnormalized log probabilities

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Cross Entropy Loss (Softmax)For Classification

$$L_i = -\log\left(\frac{e^{sy_i}}{\sum_j e^{s_j}}\right)$$

unnormalized probabilities

	unnormalized log probabilities	exp	unnormalized probabilities	normalize	probabilities
cat	**3.2**		**24.5**		**0.13**
car	5.1		164.0		0.87
frog	-1.7		0.18		0.00

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Cross Entropy Loss (Softmax)For Classification

$$L_i = -\log\left(\frac{e^{sy_i}}{\sum_j e^{s_j}}\right)$$

unnormalized probabilities

	unnormalized log probabilities	exp		normalize	probabilities	
cat	**3.2**	→	**24.5**	→	**0.13**	→ L_i = -log(0.13) = **0.89**
car	5.1	→	164.0	→	0.87	
frog	-1.7	→	0.18	→	0.00	

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

How Do We Efficiently Minimize the Loss?

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

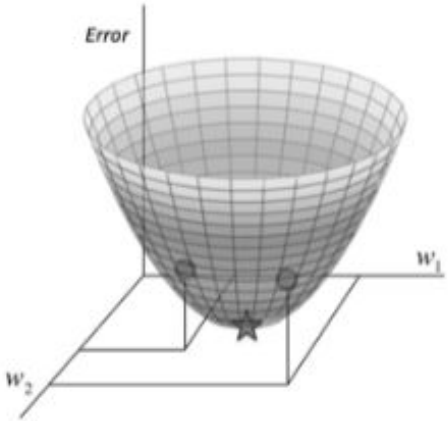
Gradient Descent Important Considerations

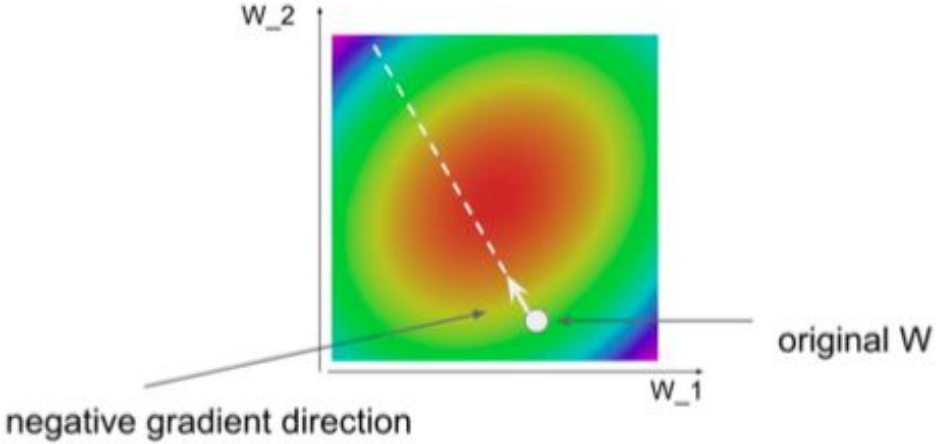
Which direction should I head?

How large or small a step?

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Gradient Descent





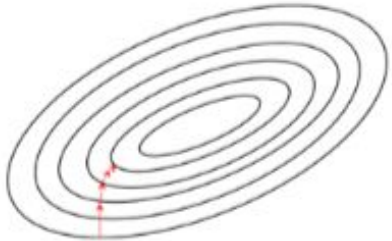
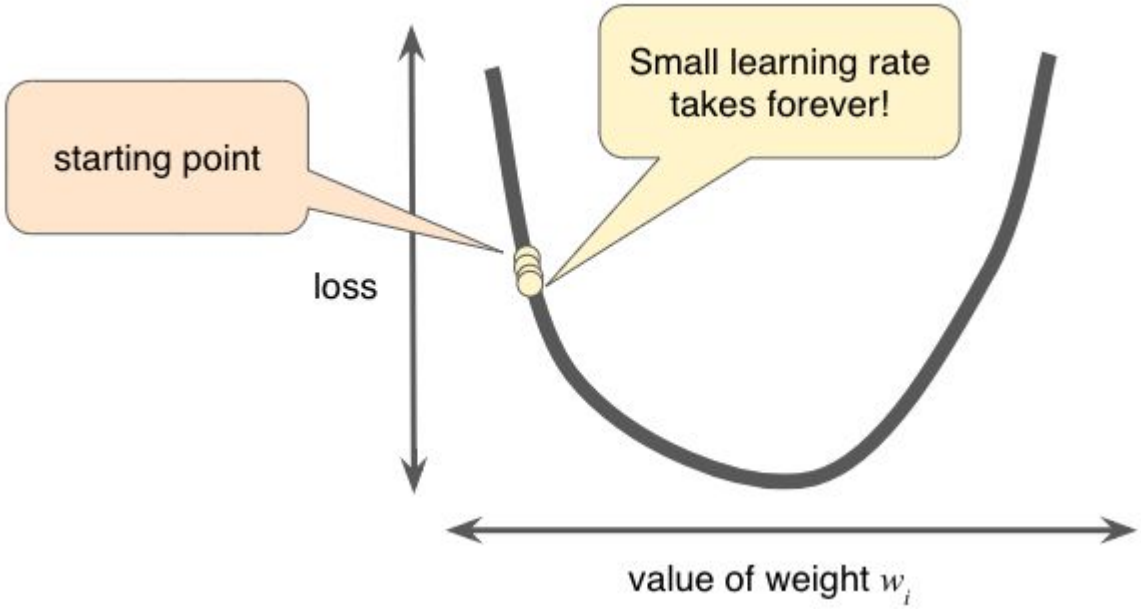


Figure 2-3. Visualizing the error surface as a set of contours

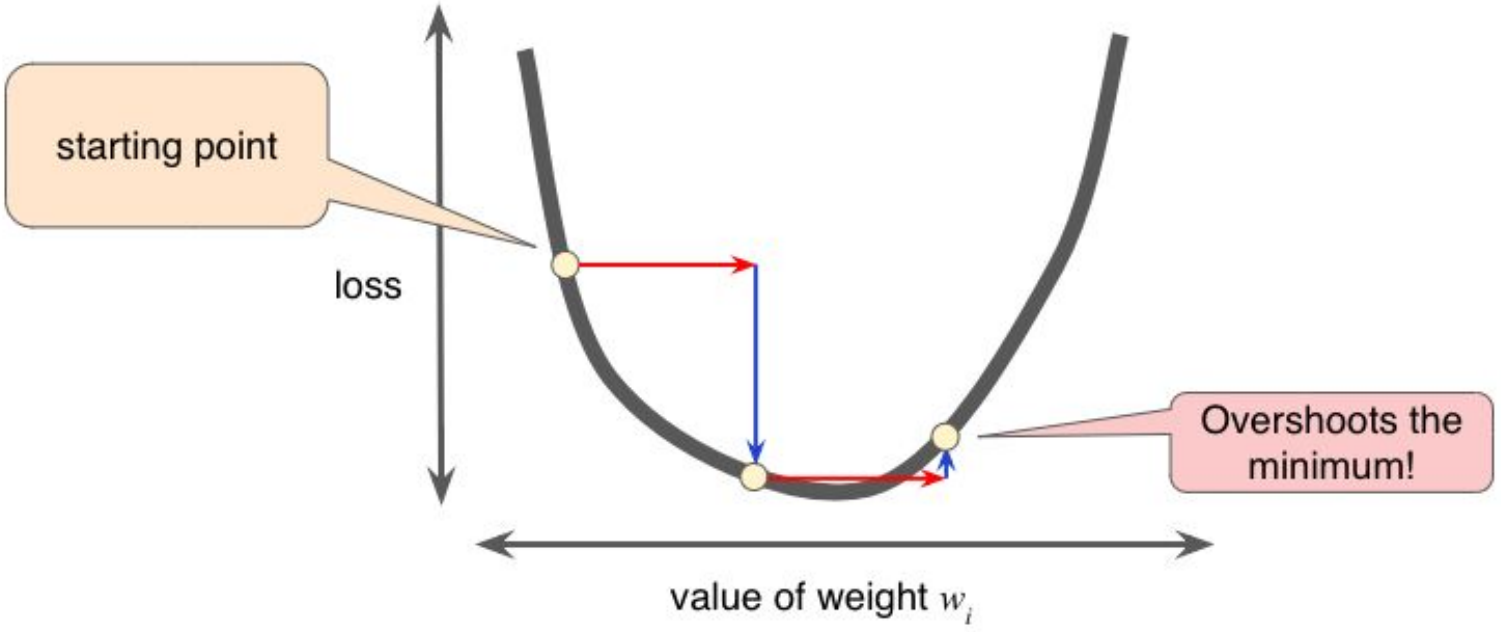
© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Learning Rate Too Small



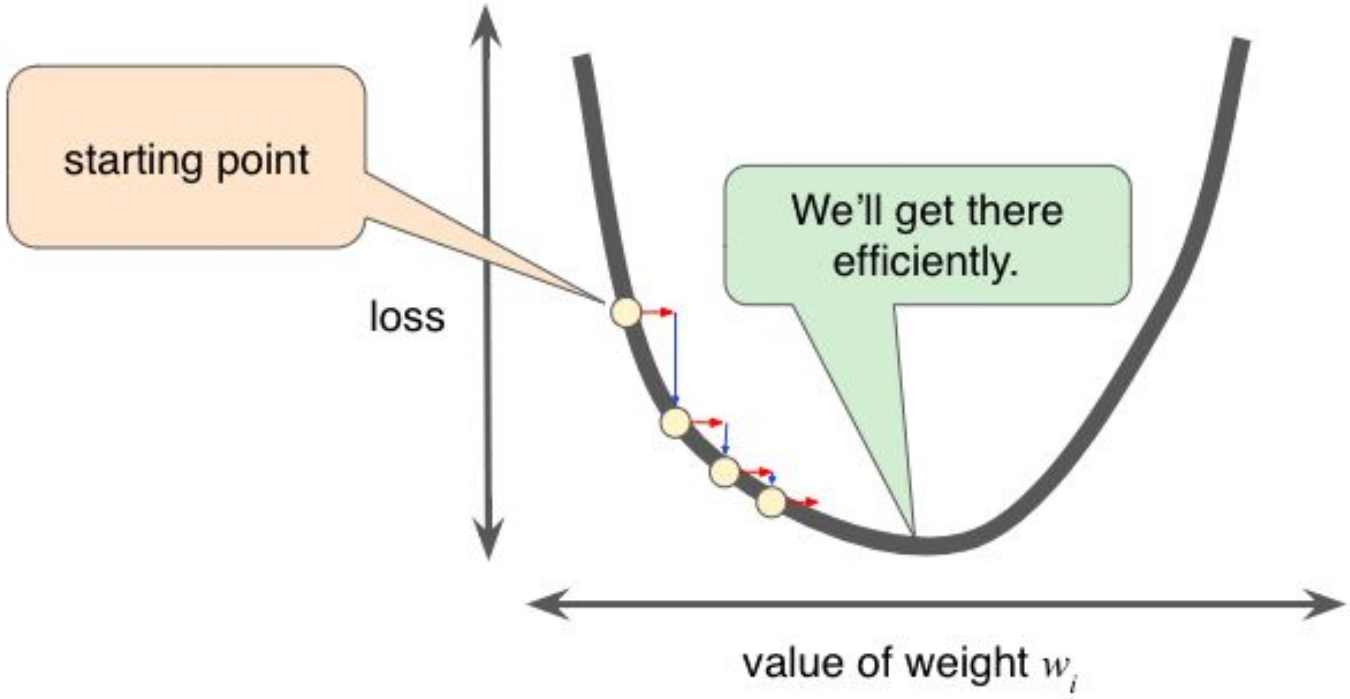
© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Learning Rate Too Large



© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Optimal Learning Rate

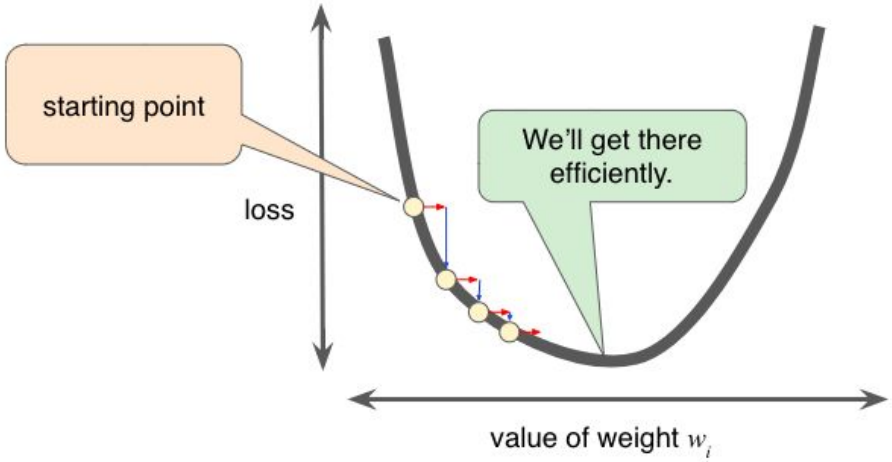


© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Gradient Descent

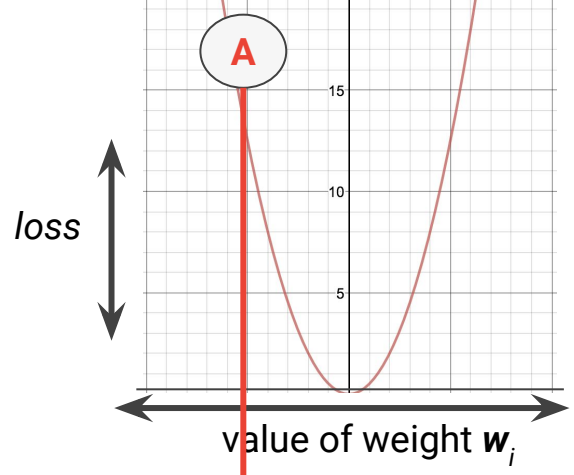
```
# Vanilla Gradient Descent

while True:
  weights_grad = evaluate_gradient(loss_fun, data, weights)
  weights += - step_size * weights_grad # perform parameter update
```

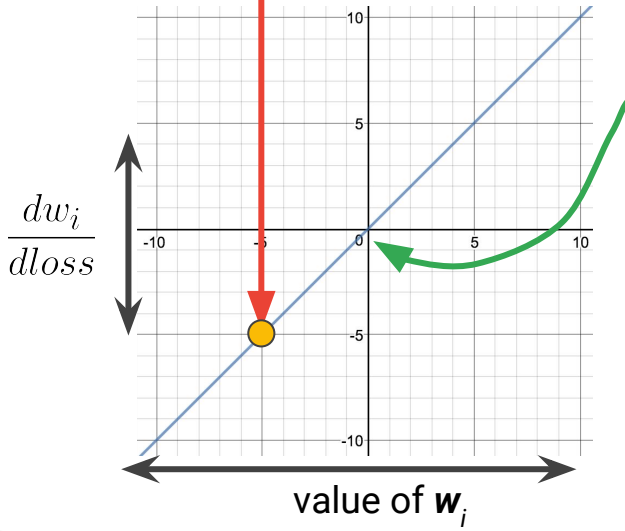


© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

The Loss Function slope provides direction and step size in our search



Loss Function
(e.g. RMSE)



Remember, our goal is to find the minimum loss which is where the Loss Function slope 0

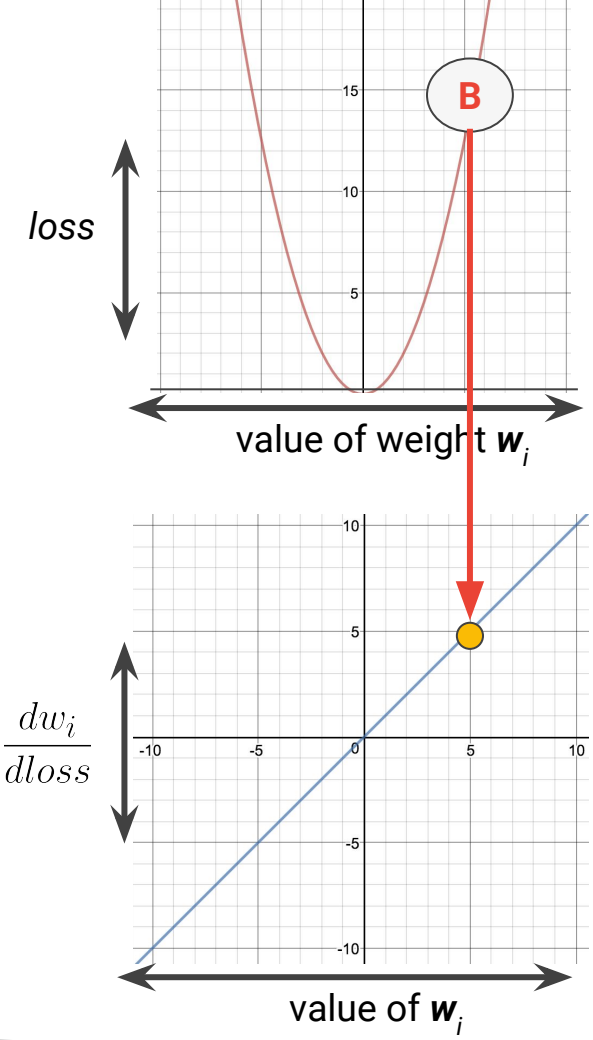
Loss Function Slope (Derivative)

Slope is Negative
Direction: Go Right!

Magnitude is (-5)
Step Size: Big

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

The Loss Function slope provides direction and step size in our search



Loss Function
(e.g. RMSE)

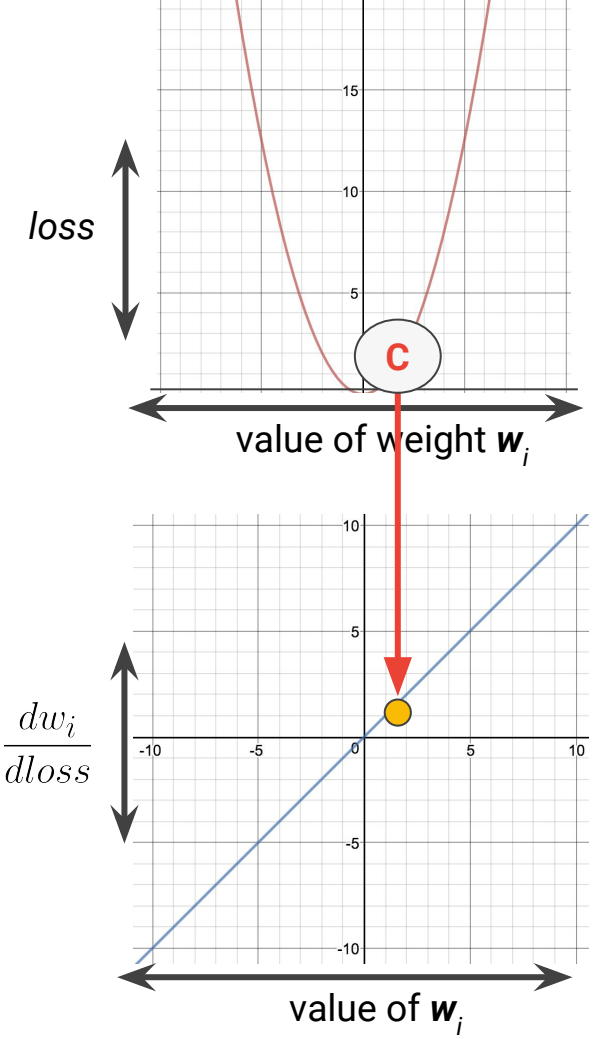
Loss Function Slope (Derivative)

Slope is Positive
Direction: Go Left!

Magnitude is 5
Step Size: Big

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

The Loss Function slope provides direction and step size in our search



Loss Function
(e.g. RMSE)

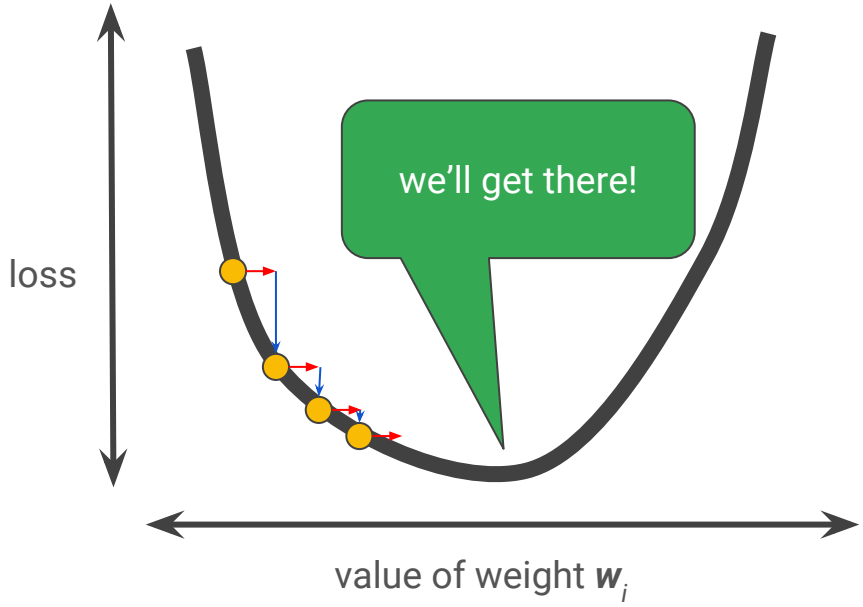
Loss Function Slope (Derivative)

Slope is Positive
Direction: Go Left!

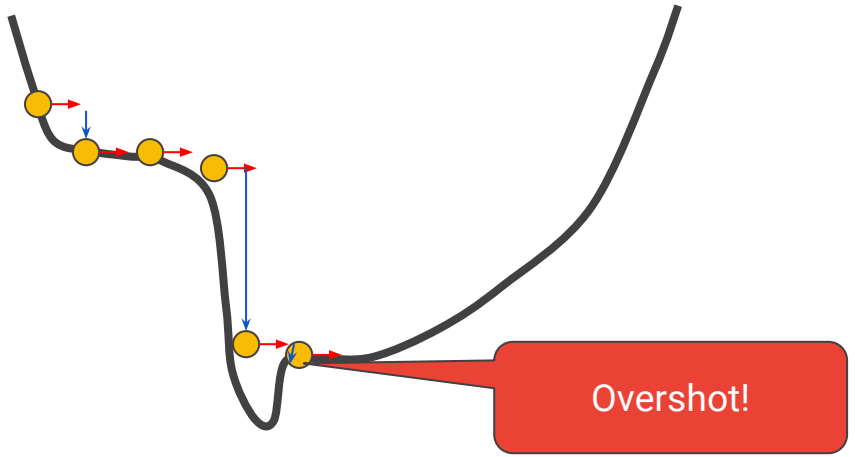
Magnitude is 2
Step Size: Small

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

A correct and constant step size can prove difficult to find



Step size or "learning rate" is a hyper-parameter which is set before training



one size does not fit all models

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

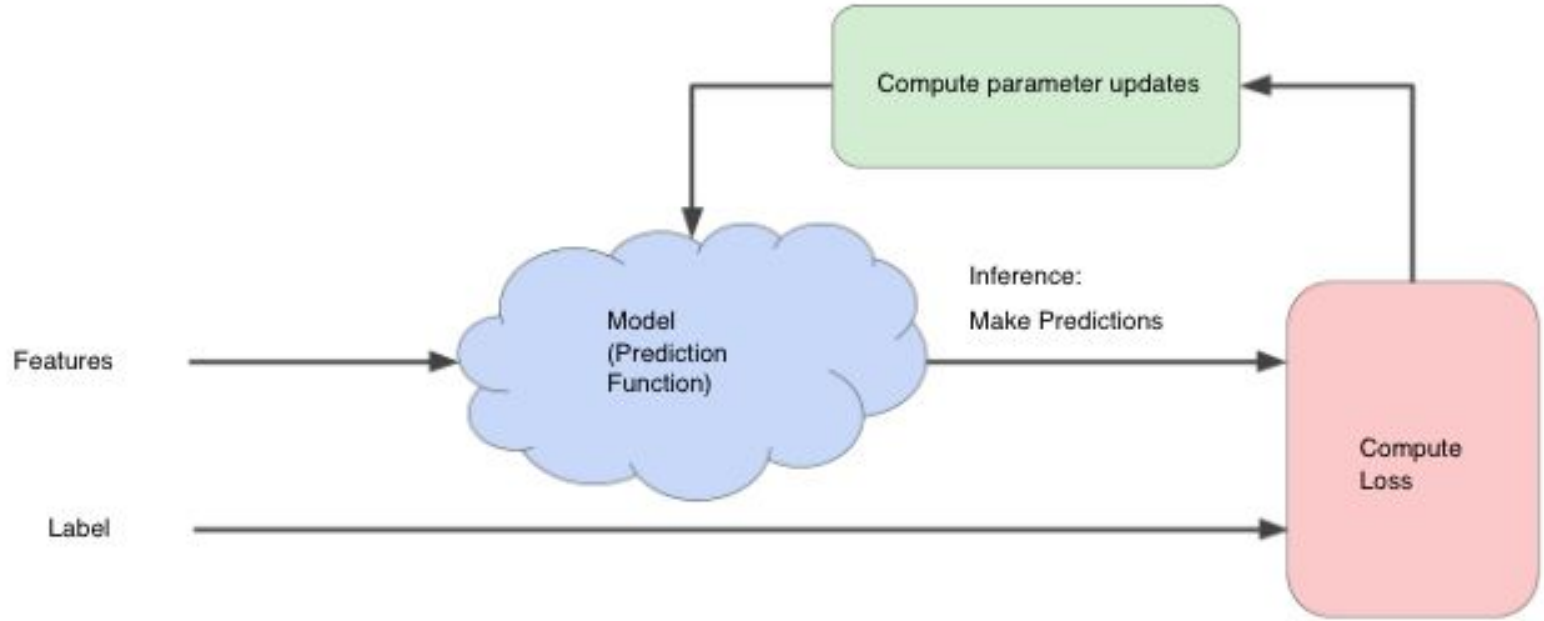
Linear Regression Quiz

What is a hyperparameter that helps determine gradient descent's step size along the hypersurface to hopefully speed up convergence?

A. Collinearity measure
B. Learning rate
C. Gram matrix discriminant
D. Condition number

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Iterative Weight Update Process



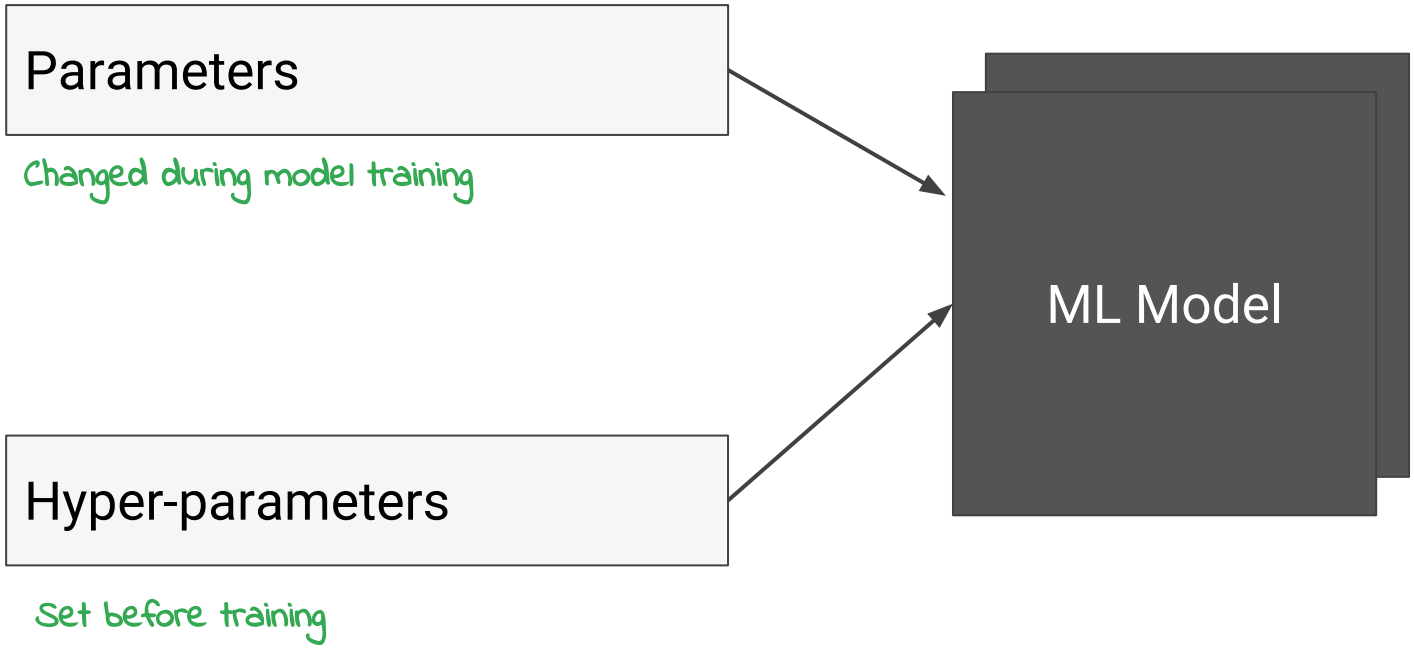
© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Agenda

- Supervised and Unsupervised Machine Learning
- Regression and Classification
- Loss and Gradient Descent
- Hyperparameters (learning rate, batch size)
- Accuracy Metrics
- Preventing Overfitting
- Intro to Neural Nets

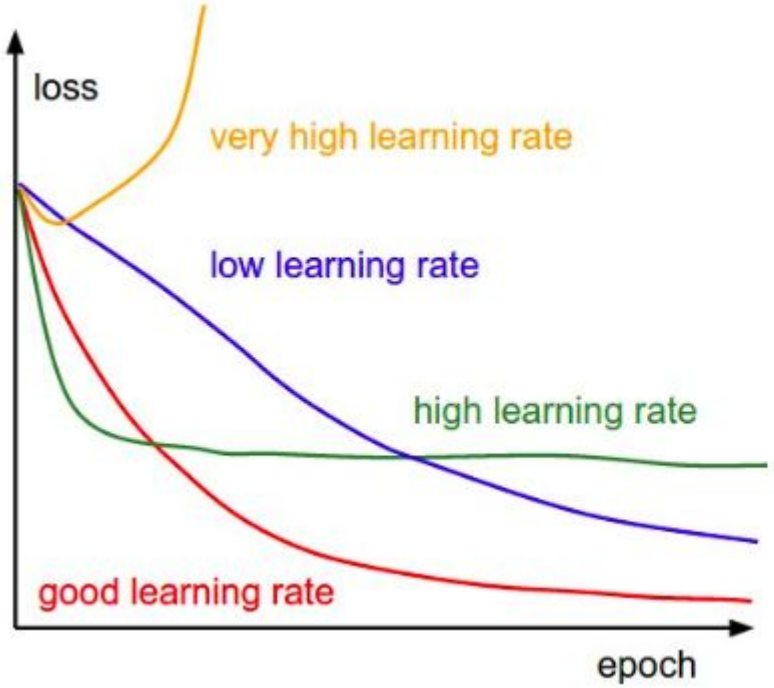
© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

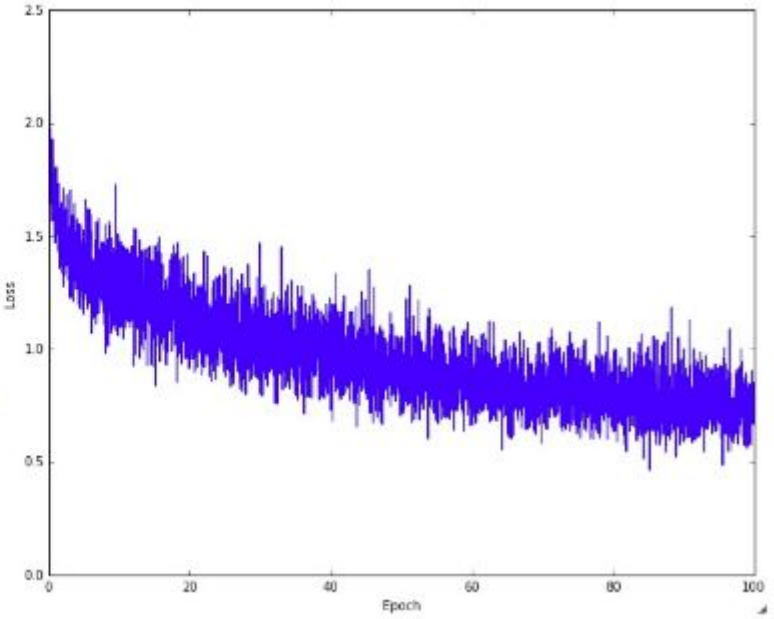
ML models are mathematical functions with parameters and hyper-parameters



© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Learning Rate





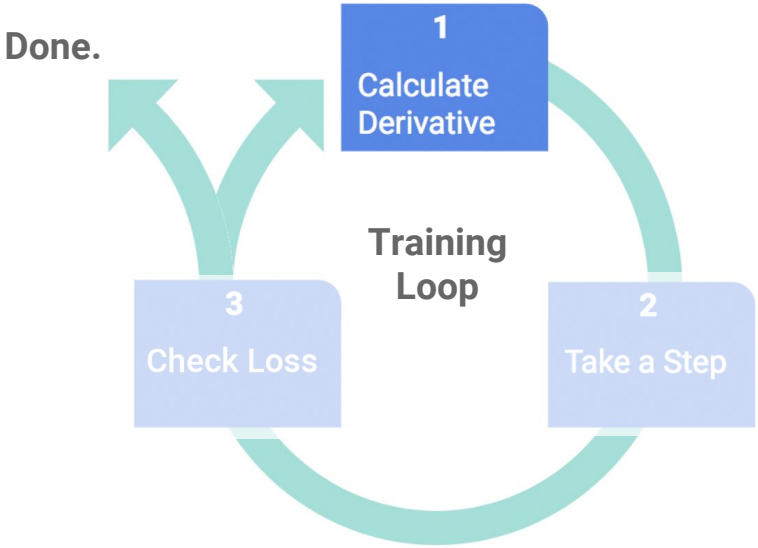
© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Learning Rate Demo

https://developers.google.com/machine-learning/crash-course/fitter/graph

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

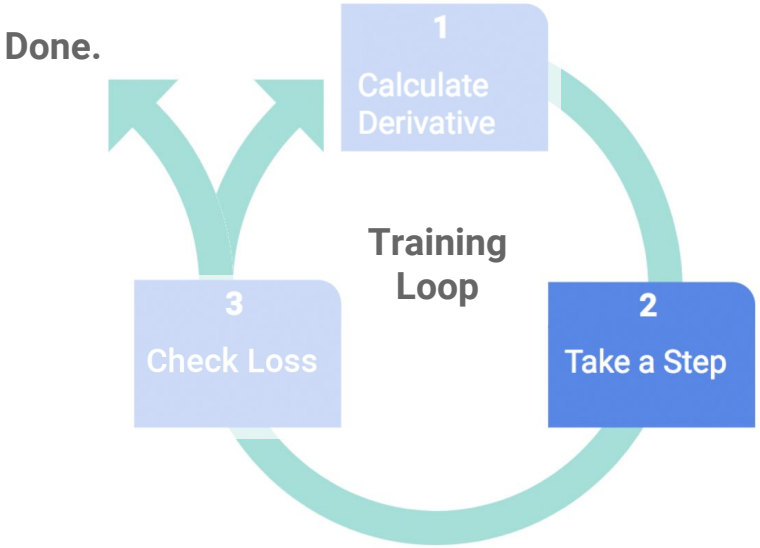
Problem: Model training is still too slow



- Number of data points
- Number of model parameters

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

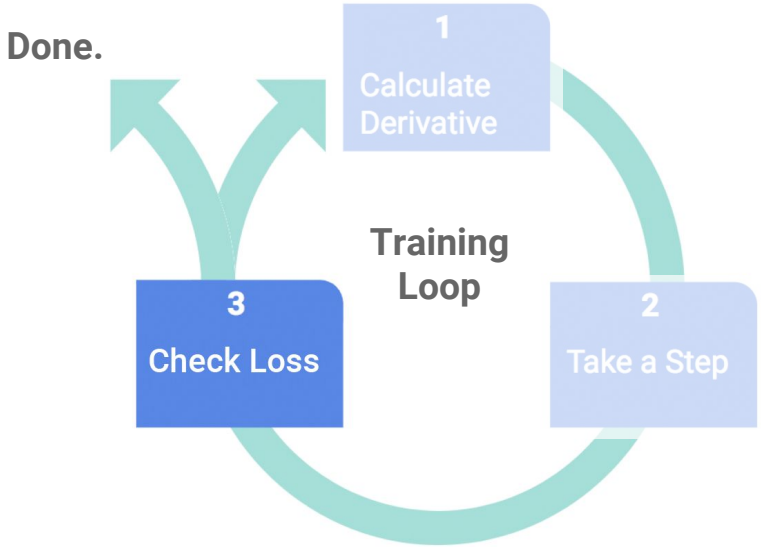
Problem: Model training is still too slow



- Number of model parameters

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

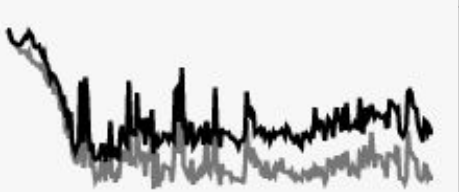
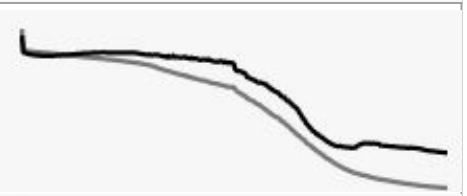
Problem: Model training is still too slow



- Number of data points
- Number of model parameters
- Frequency

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

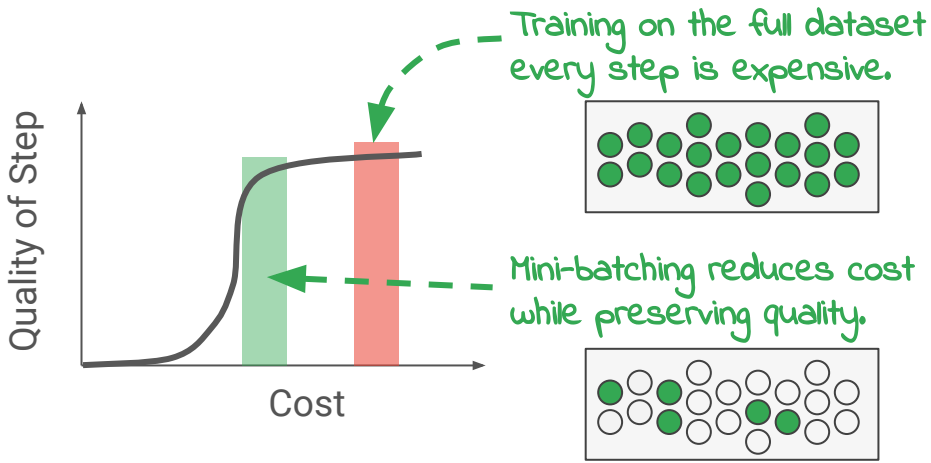
Batch Size

Batch Size	Loss Over Time
1	
10	
30	

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Calculating the derivative on fewer data points

Mini Batch Stochastic Gradient Descent (SGD)



Typical values for batch size: 10 - 1000 examples.

```
# Vanilla Minibatch Gradient Descent

while True:
  data_batch = sample_training_data(data, 256) # sample 256 examples
  weights_grad = evaluate_gradient(loss_fun, data_batch, weights)
  weights += - step_size * weights_grad # perform parameter update
```

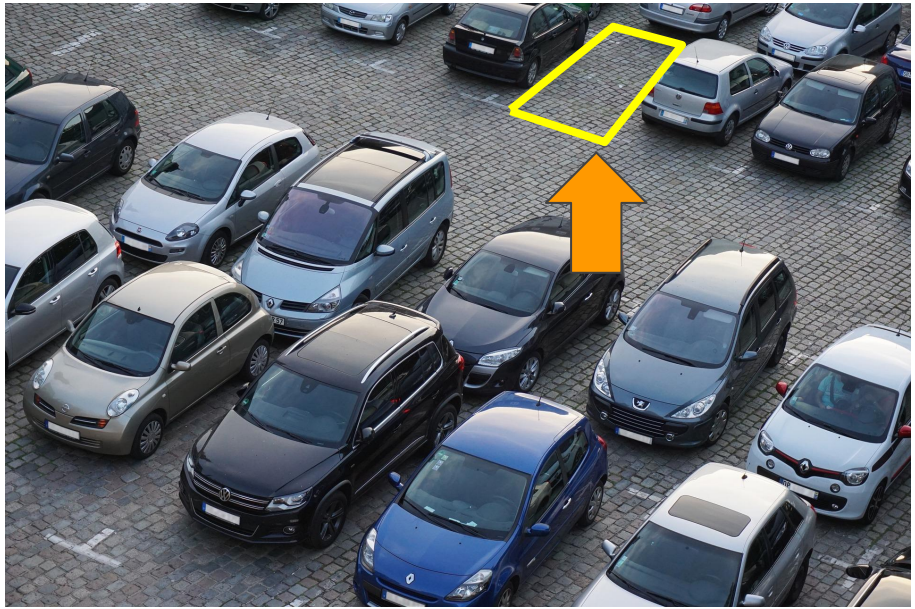
© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Agenda

- Supervised and Unsupervised Machine Learning
- Regression and Classification
- Loss and Gradient Descent
- Hyperparameters (learning rate, batch size)
- Accuracy Metrics
- Preventing Overfitting
- Intro to Neural Nets

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Skewed data can make inappropriate strategies seductive



1000 parking spaces
990 of them are **taken**
10 are **available**

An ML model that always reported that a space was occupied would be right 99/100 times.

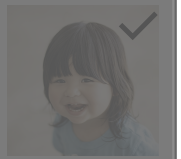
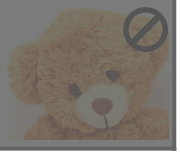
© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Performance metrics allow us to measure what matters

Loss Functions	Performance Metrics
• During training • Harder to understand • Indirectly connected to business goals	• After training • Easier to understand • Directly connected to business goals

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Use a confusion matrix to assess classification model performance

Labels	Model Predictions: Positive	Model Predictions: Negative
Positive	True Positives (TP) 	False Negatives (FN) *Type II Error* 
Negative	False Positives (FP) *Type I Error* 	True Negatives (TN) 

Model says Yes

Model says No

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

True positive: Correctly predict parking spot is available

References	Model Predictions: Positive	Model Predictions: Negative
Positive	Available parking space exists Model predicts it is available **True Positives**	Available parking space exists Model doesn't predict it **False Negatives** *Type II Error*
Negative	Available parking space doesn't exist Model predicts it is available **False Positives** *Type I error*	Available parking space doesn't exist Model correctly doesn't predict it **True Negatives**

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

False positive: Predicting a spot is there when it is not

		Model Predictions: Positive	Model Predictions: Negative
References	Positive	Available parking space exists Model predicts it is available **True Positives**	Available parking space exists Model doesn't predict it **False Negatives** *Type II Error*
References	Negative	Available parking space doesn't exist Model predicts it is available **False Positives** *Type I error*	Available parking space doesn't exist Model correctly doesn't predict it **True Negatives**

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

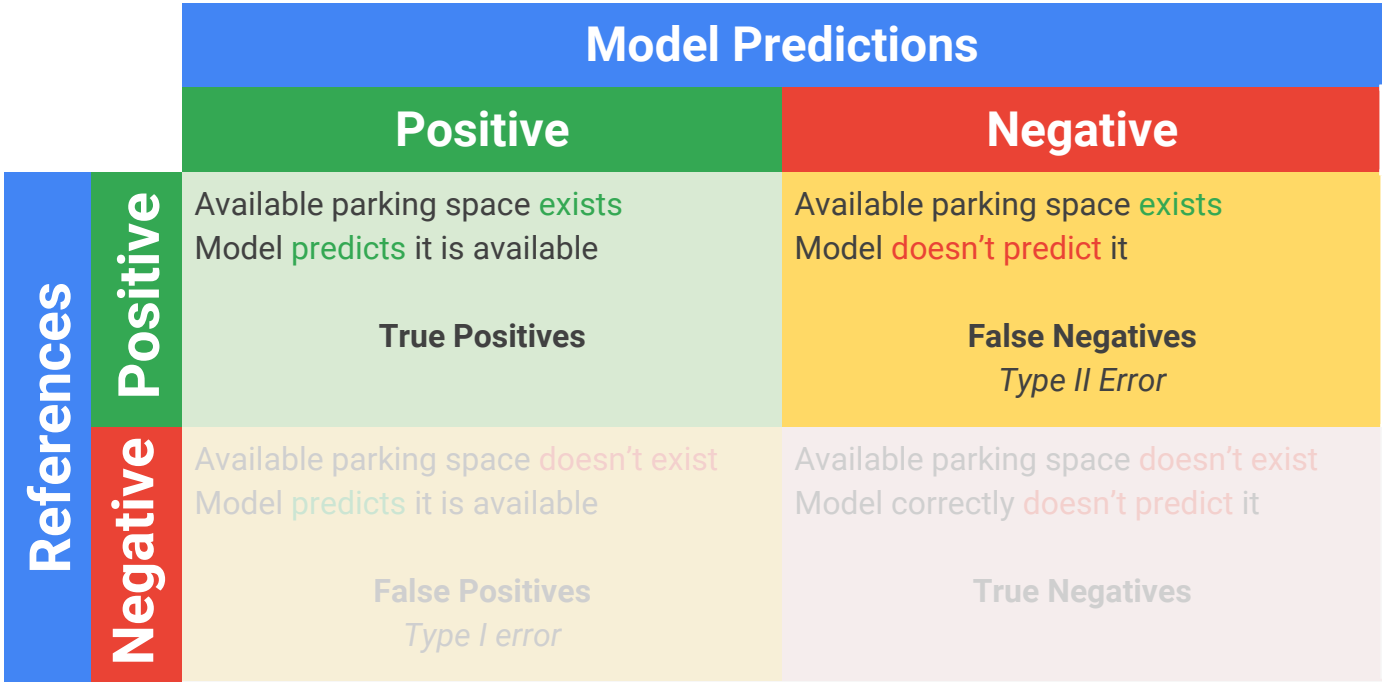
Precision: true positives / total classified as positive

		Model Predictions: Positive	Model Predictions: Negative
References	Positive	Available parking space exists Model predicts it is available **True Positives**	Available parking space exists Model doesn't predict it **False Negatives** *Type II Error*
References	Negative	Available parking space doesn't exist Model predicts it is available **False Positives** *Type I error*	Available parking space doesn't exist Model correctly doesn't predict it **True Negatives**

Precision

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Recall: true positives / all actual positives in our reference



Recall

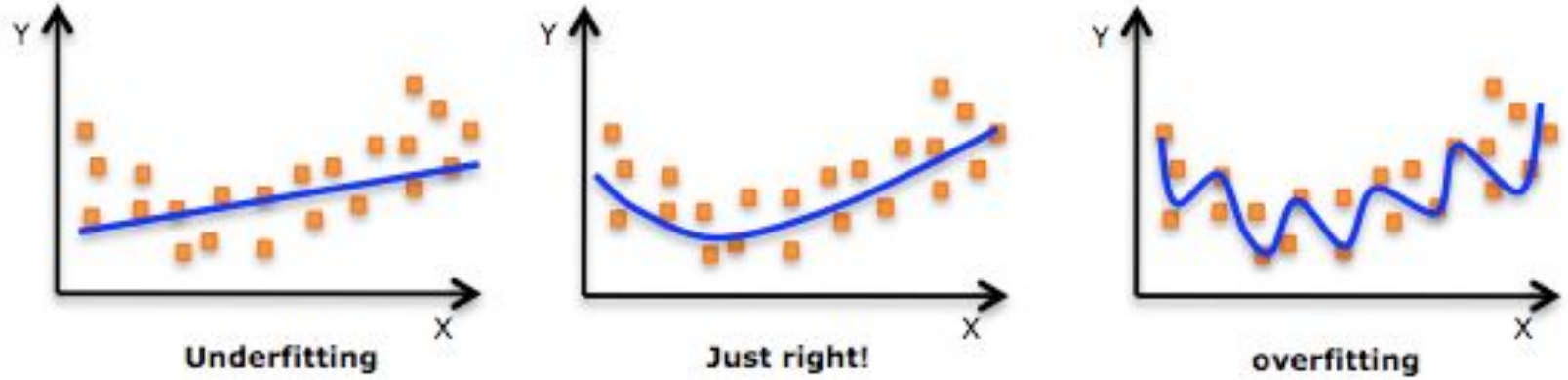
© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Agenda

- Supervised and Unsupervised Machine Learning
- Regression and Classification
- Loss and Gradient Descent
- Hyperparameters (learning rate, batch size)
- Accuracy Metrics
- Preventing Overfitting
- Intro to Neural Nets

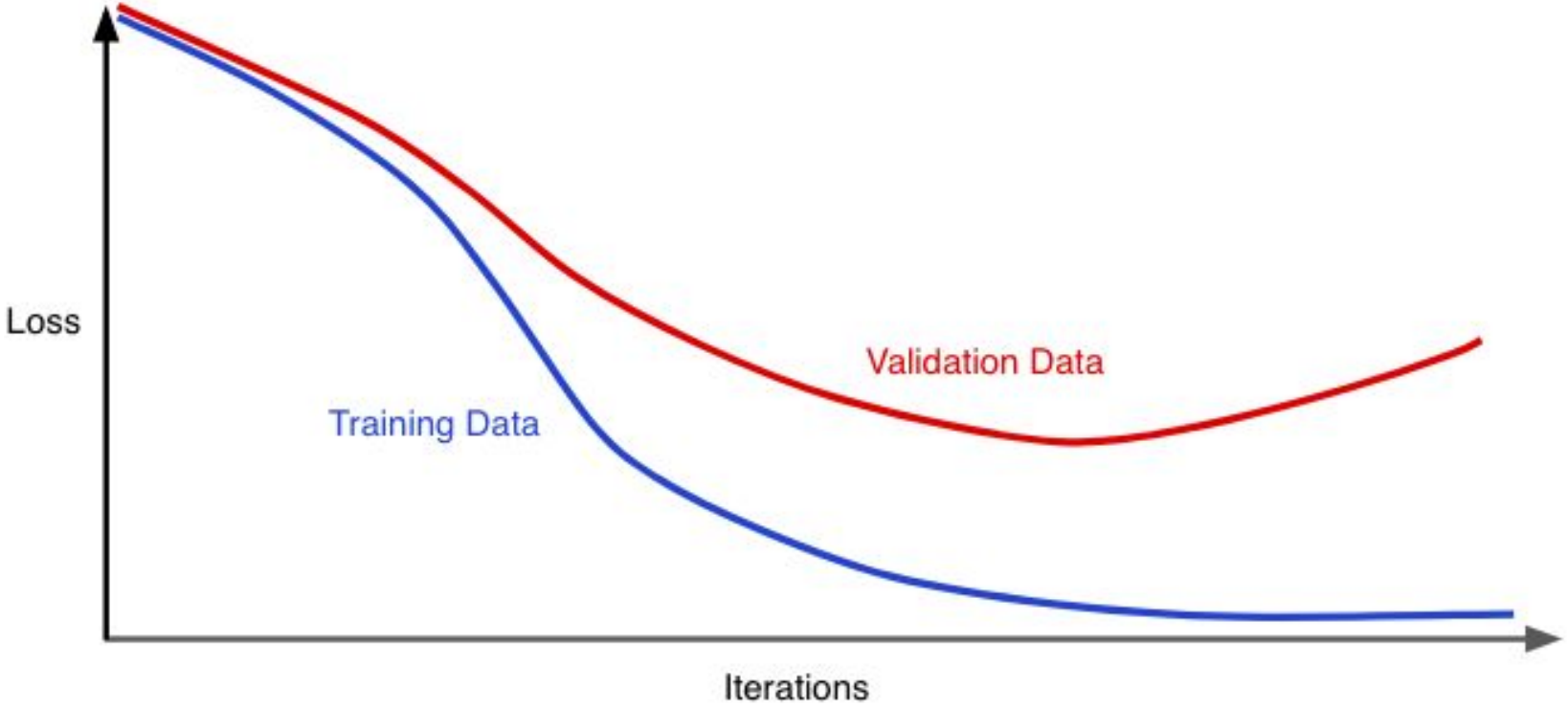
© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Under/Overfitting



- Training Set, Cross Validation Set, and Test Set
- Early stopping
- Varied examples
- Regularization

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

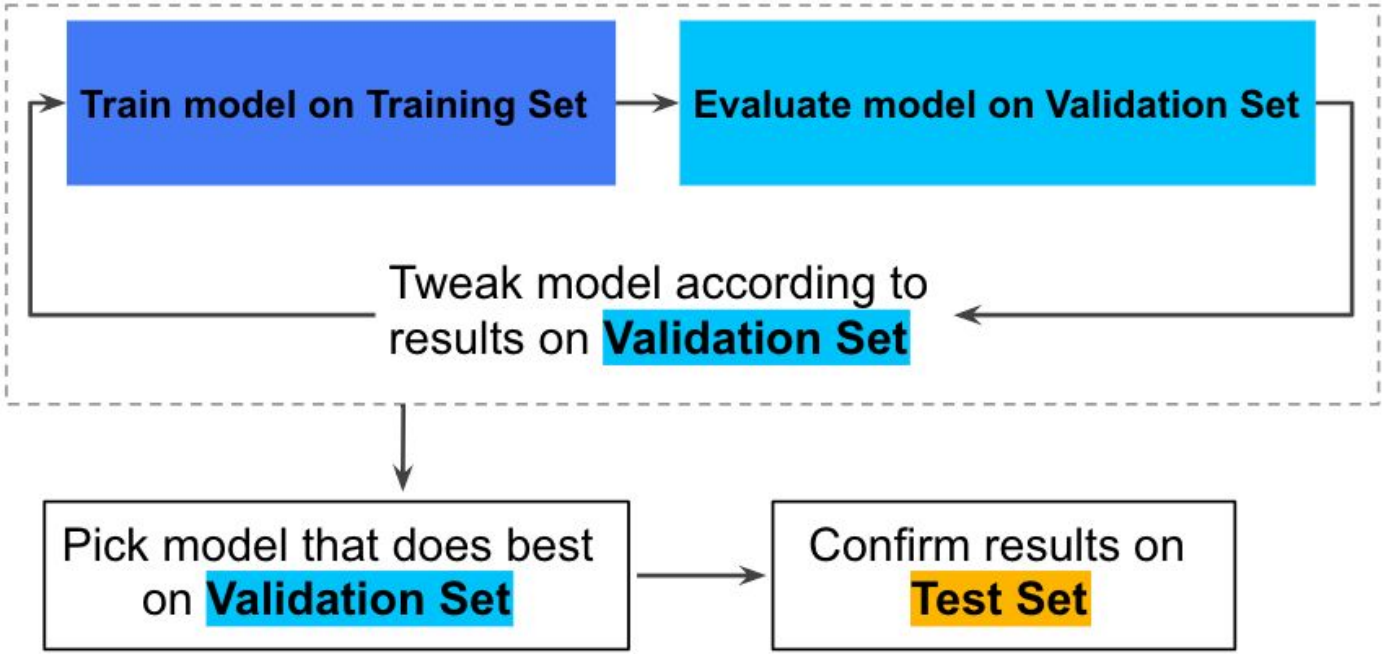


© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Evaluate Fit By Splitting Up Your Data Into Training, Validation, and Test Sets

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Evaluate Fit By Splitting Up Your Data Into Training, Validation, and Test Sets



© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

L2 Regularization

If model complexity is a function of weights, a feature weight with a high absolute value is more complex than a feature weight with a low absolute value.

the ***L_2* regularization** formula, which defines the regularization term as the sum of the squares of all the feature weights:

$$L_2 \text{ regularization term} = ||\boldsymbol{w}||_2^2 = w_1^2 + w_2^2 + \ldots + w_n^2$$

$$\text{minimize}(\text{Loss}(\text{Data}|\text{Model}) + \lambda \, \text{complexity}(\text{Model}))$$

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

L1 Regularization

In a high-dimensional sparse vector, it would be nice to encourage weights to drop to exactly `0` where possible.

L_2 regularization encourages weights to be small, but doesn't force them to exactly 0.0...

We can use L_1 regularization to encourage many of the uninformative coefficients in our model to be exactly 0, and thus reap RAM savings at inference time.

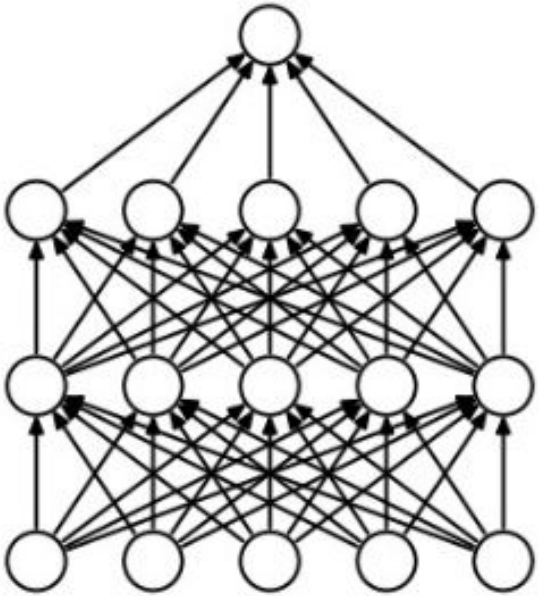
L_2 and L_1 penalize weights differently:

- L_2 penalizes $weight^2$.
- L_1 penalizes $|weight|$.

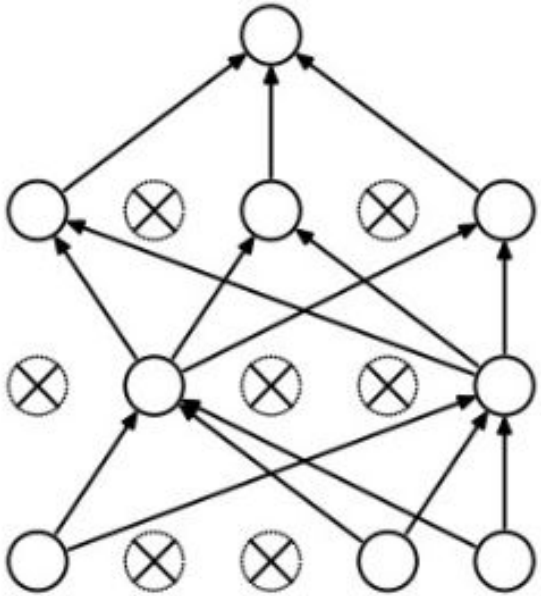
Demo:
https://developers.google.com/machine-learning/crash-course/regularization-for-sparsity/l1-regularization

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Dropout



(a) Standard Neural Net



(b) After applying dropout.

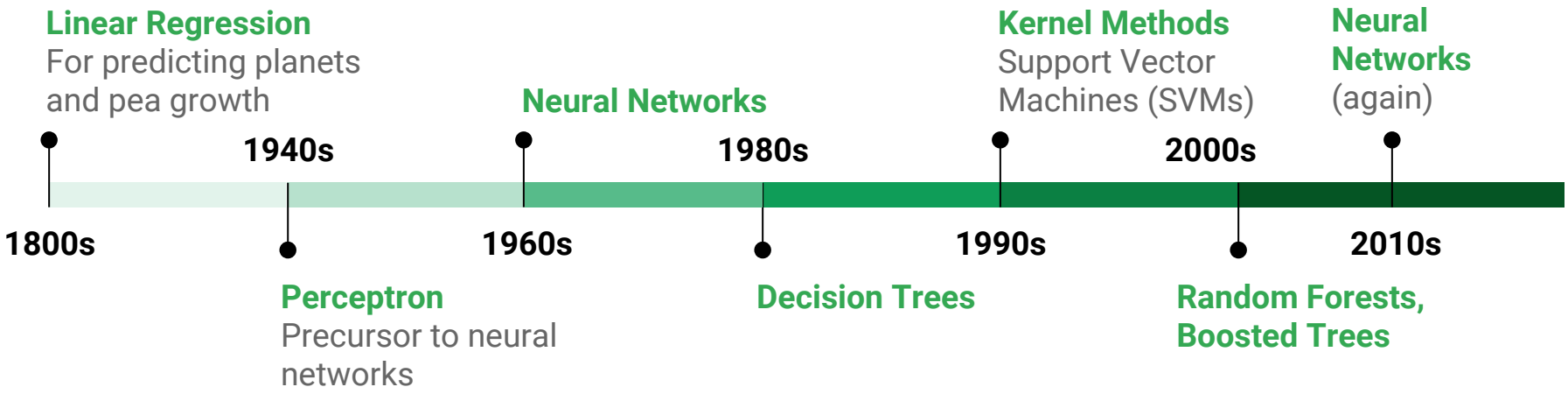
© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Agenda

- Supervised and Unsupervised Machine Learning
- Regression and Classification
- Loss and Gradient Descent
- Hyperparameters (learning rate, batch size)
- Accuracy Metrics
- Preventing Overfitting
- Intro to Neural Nets

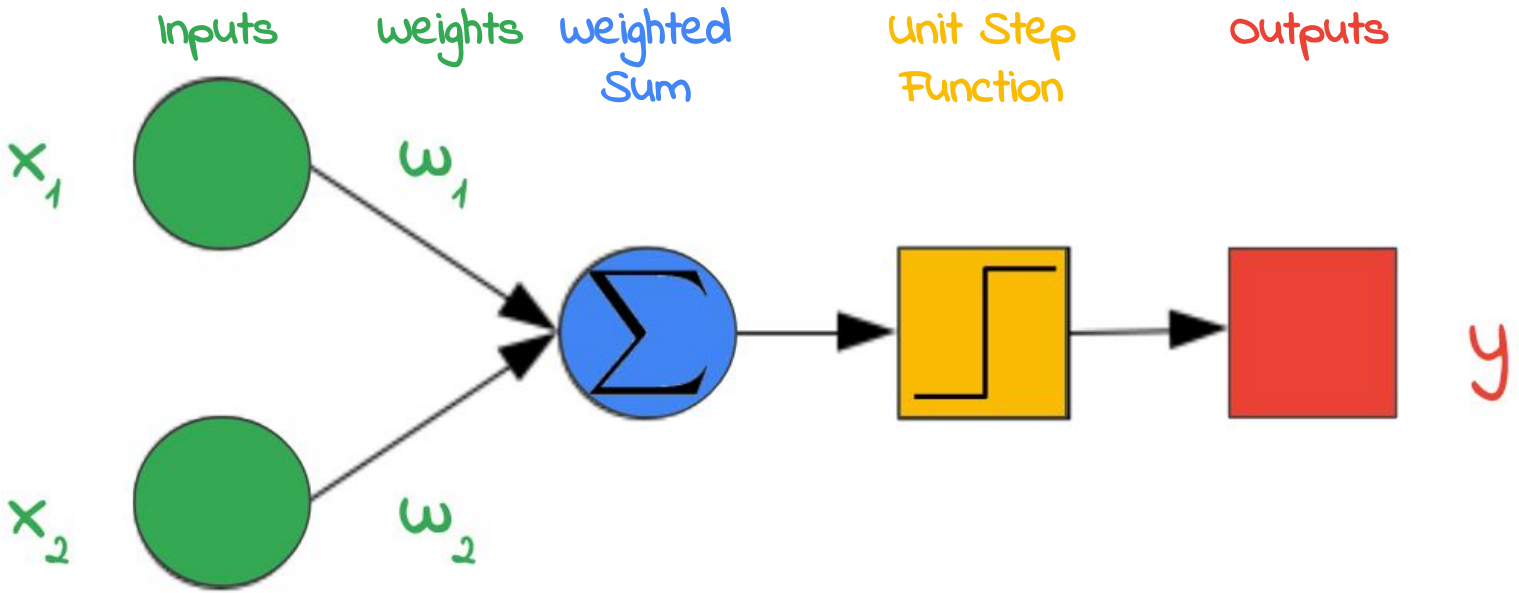
© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

With the advantage of technical improvements, more data, and computational power, neural networks made a comeback



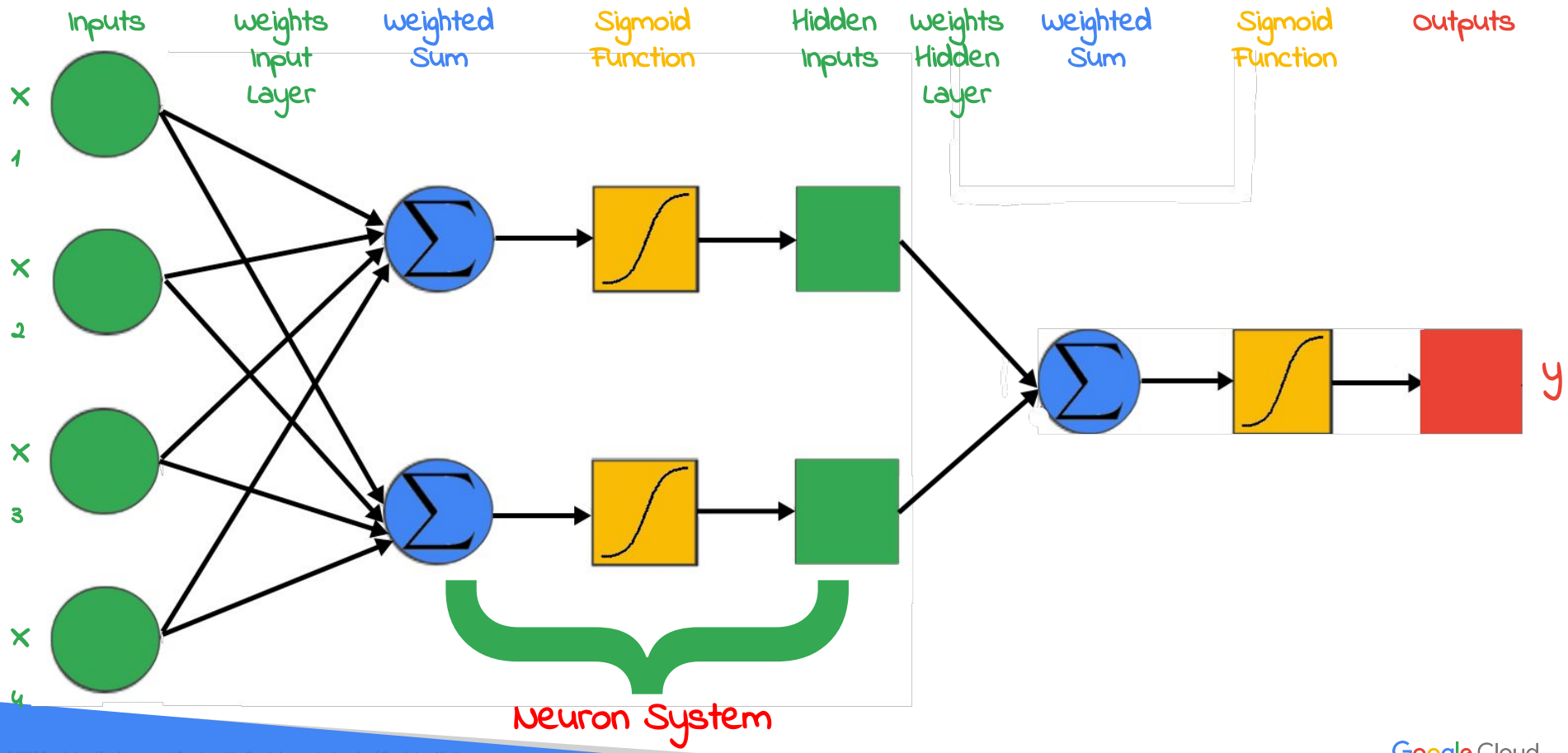
© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

The single layer perceptron was modeled on the human brain, but there are simple functions (like XOR) that it cannot learn



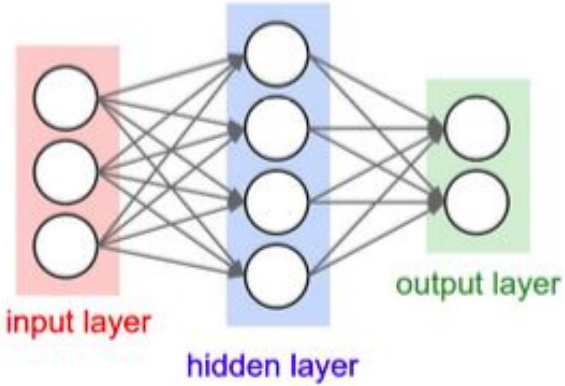
© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

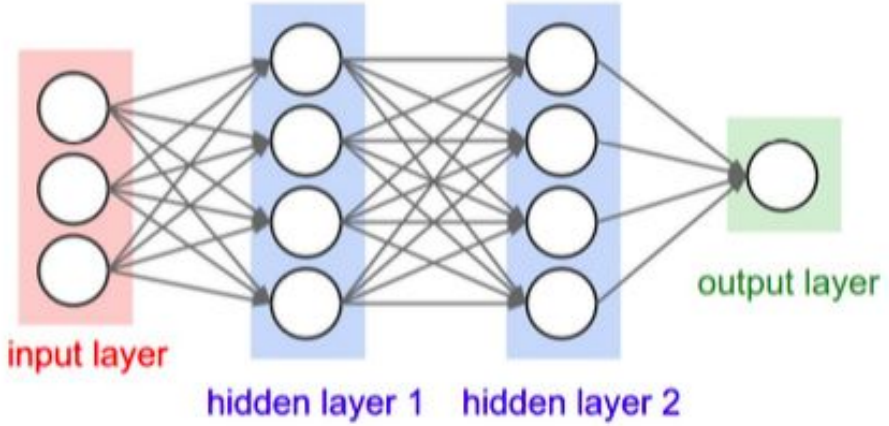
Neural networks: Multi-layer perceptron



© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

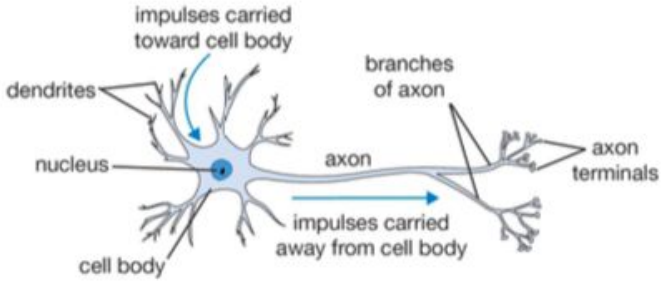
Hidden Layers

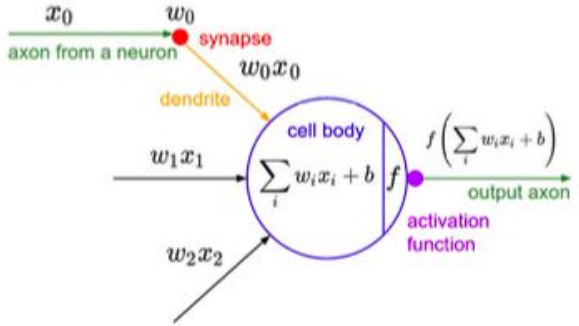




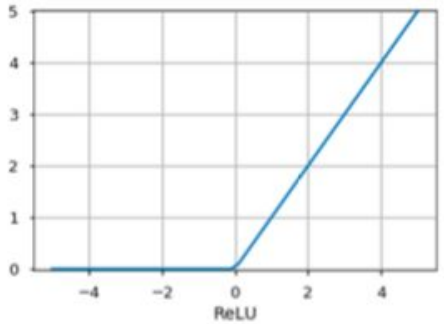
© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Similarity to the Human Brain



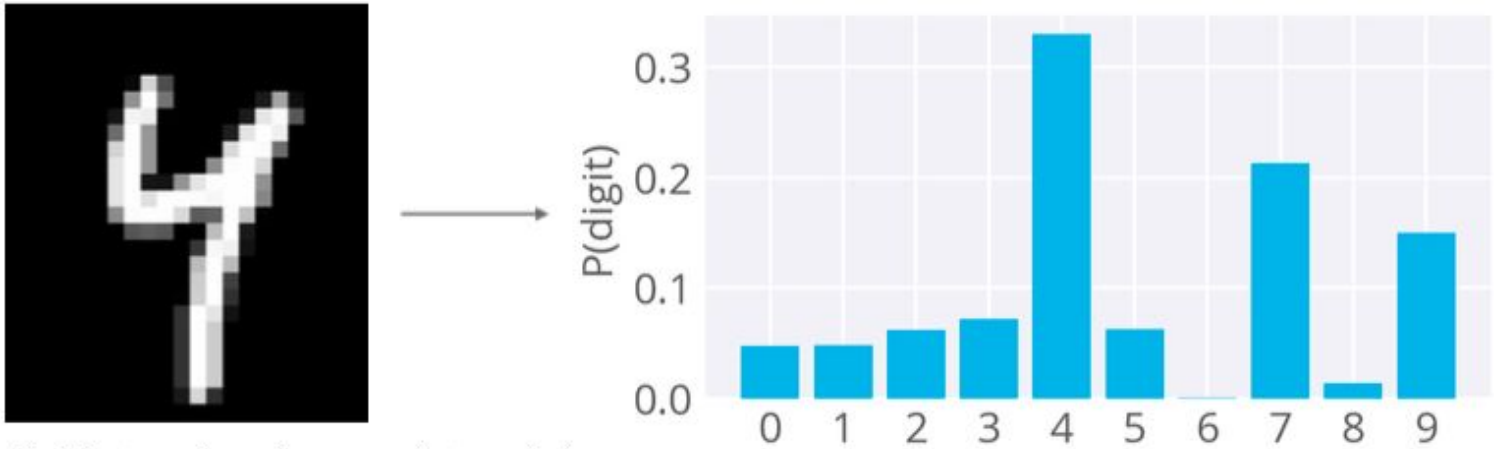


$$f(x) = max(x, 0)$$



© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Output Nodes Must Match Classes

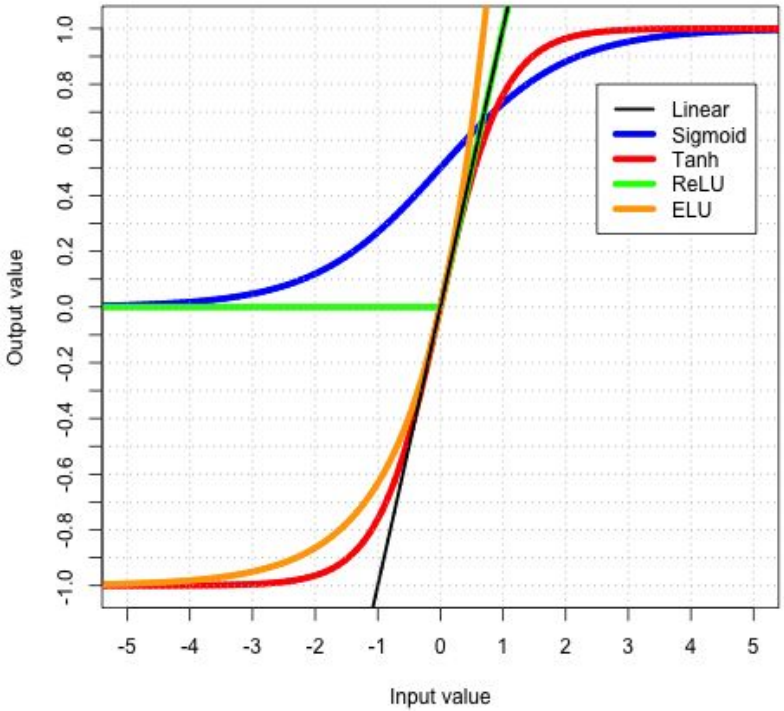


We find the true class using a one-hot encoded vector (y, our "class labels"), then take the negative log of the Softmax output for the true class. This will be our cross entropy loss:

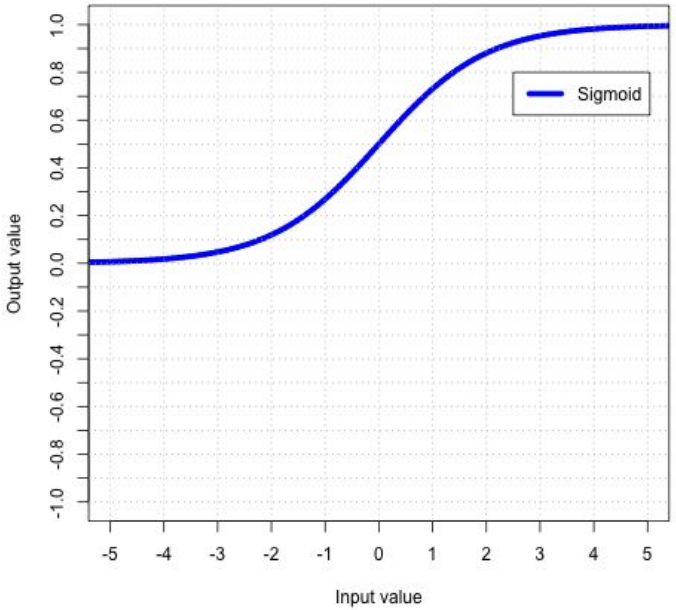
Label	0	1	2	3	4	5	6	7	8	9
Array	[0,	0,	0,	0,	1,	0,	0,	0,	0,	0]

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Neural network: Common activation functions

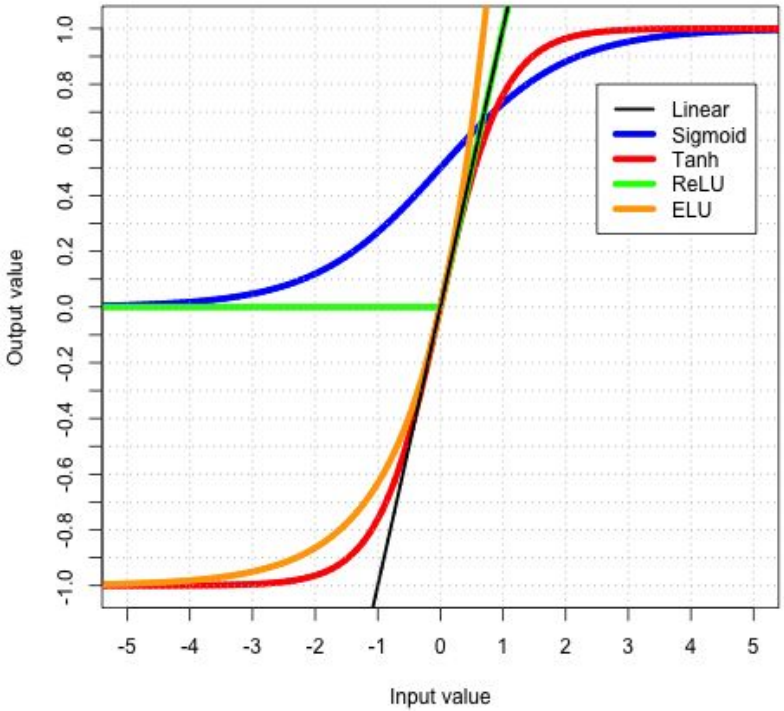


$$\text{Sigmoid} = \sigma(x) = \frac{1}{1 + e^{-x}}$$

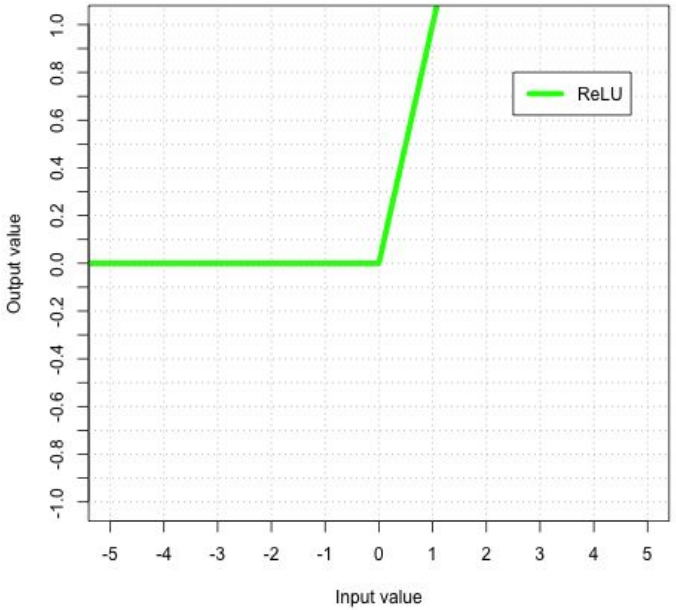


© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Neural network: Common activation functions



$$ReLU = f(x) = \begin{cases} 0 & for\ x < 0 \\ x & for\ x \geq 0 \end{cases}$$



© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

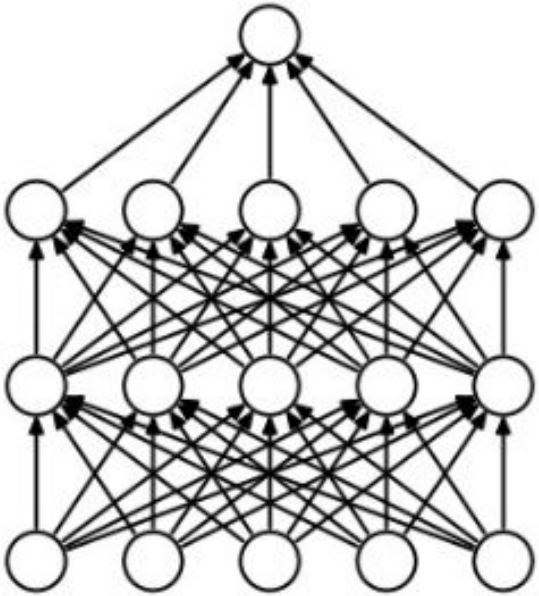
Quick Quiz!

If I wanted my outputs to be in the form of probabilities which activation function should I use in the final layer?

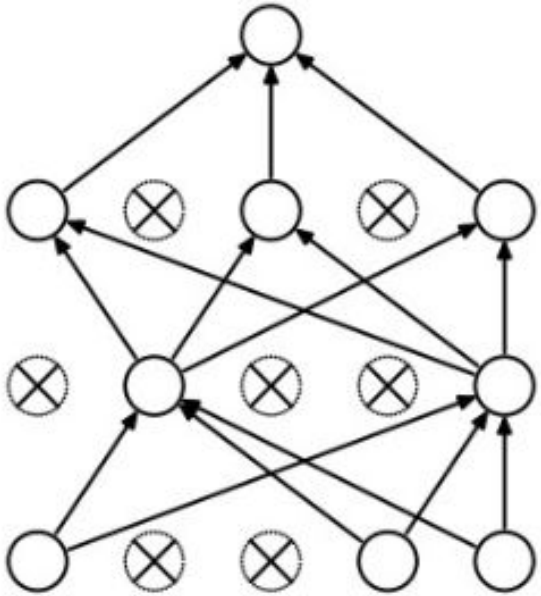
A. Sigmoid
B. Tanh
C. ReLU
D. ELU

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Dropout



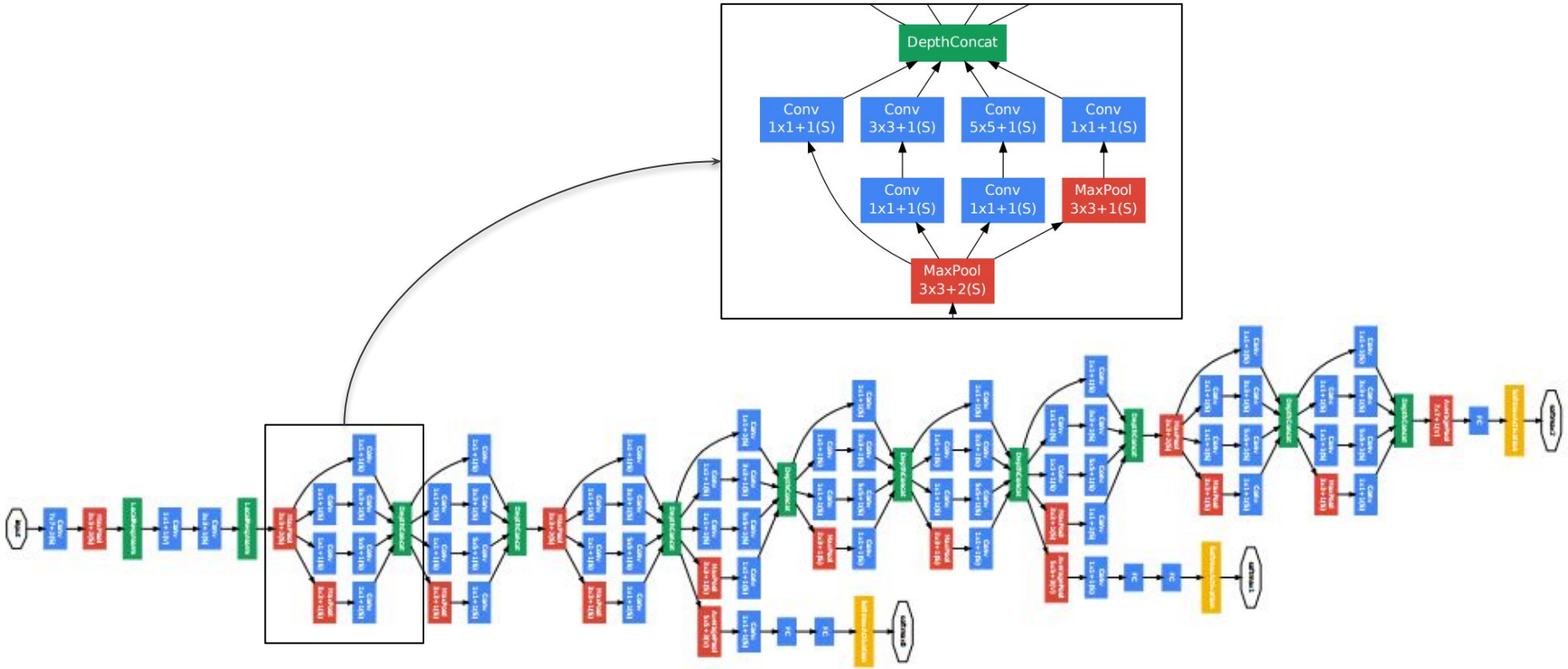
(a) Standard Neural Net



(b) After applying dropout.

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

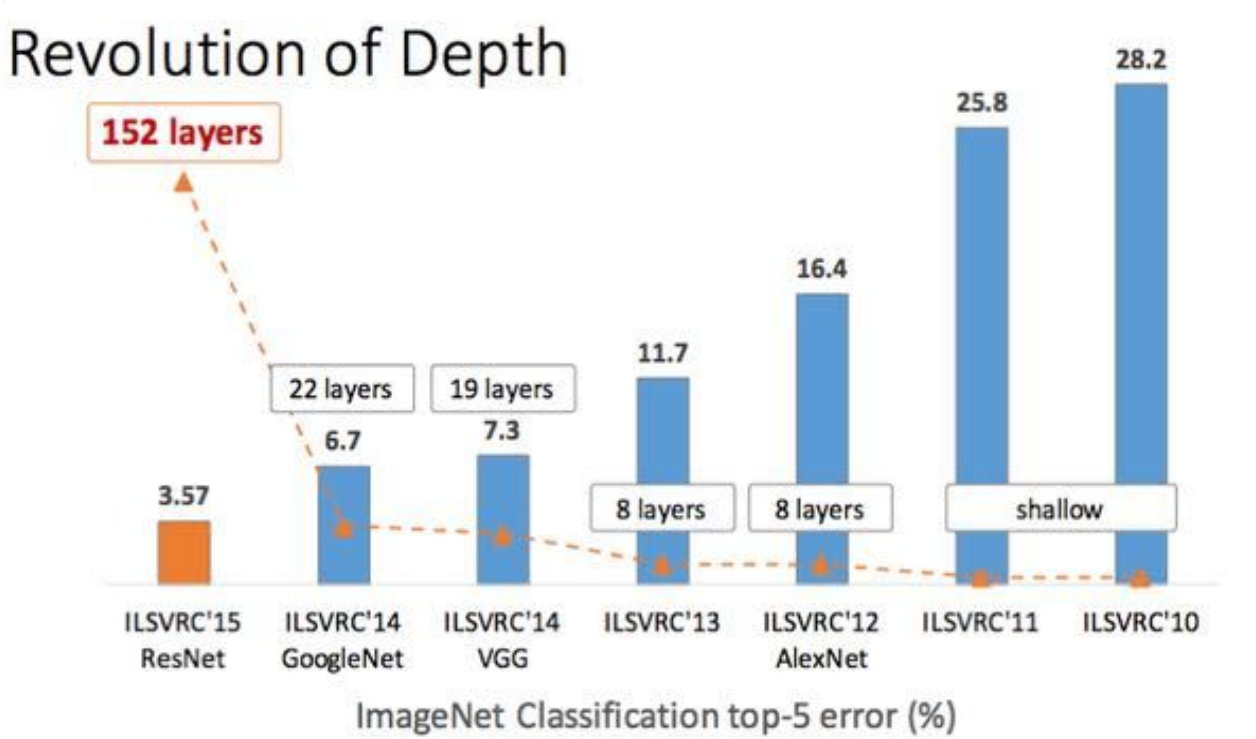
Inception/GoogLeNet Deep Neural Network



© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Google Cloud

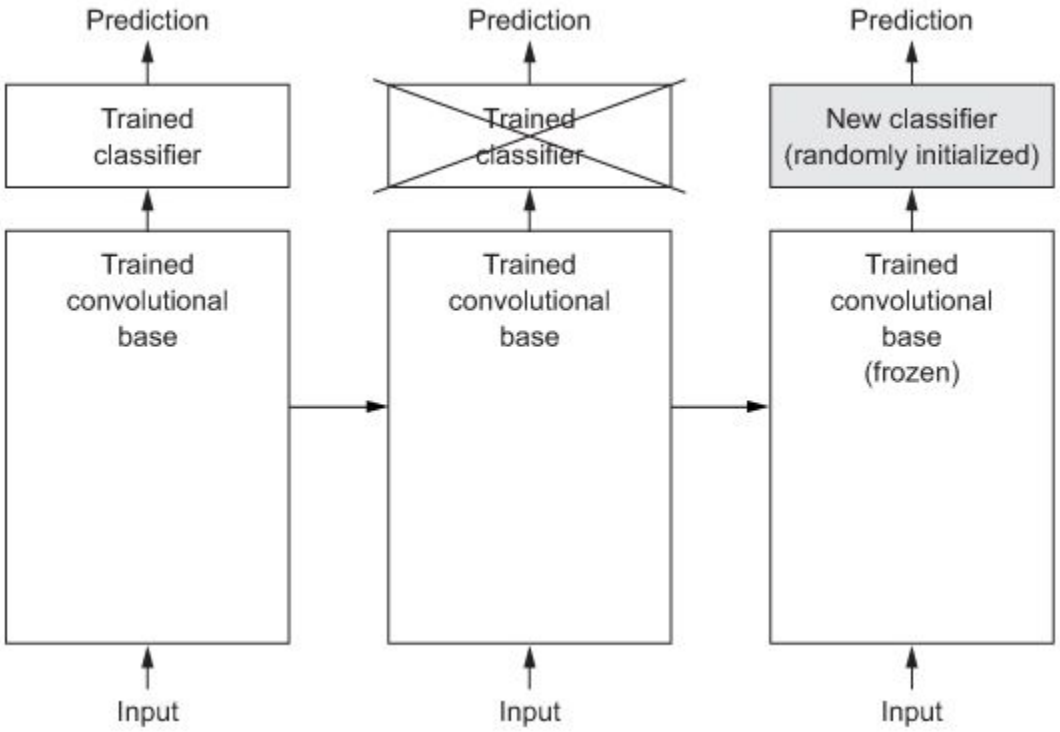
ImageNet Competition



Slide credit: Kaiming He

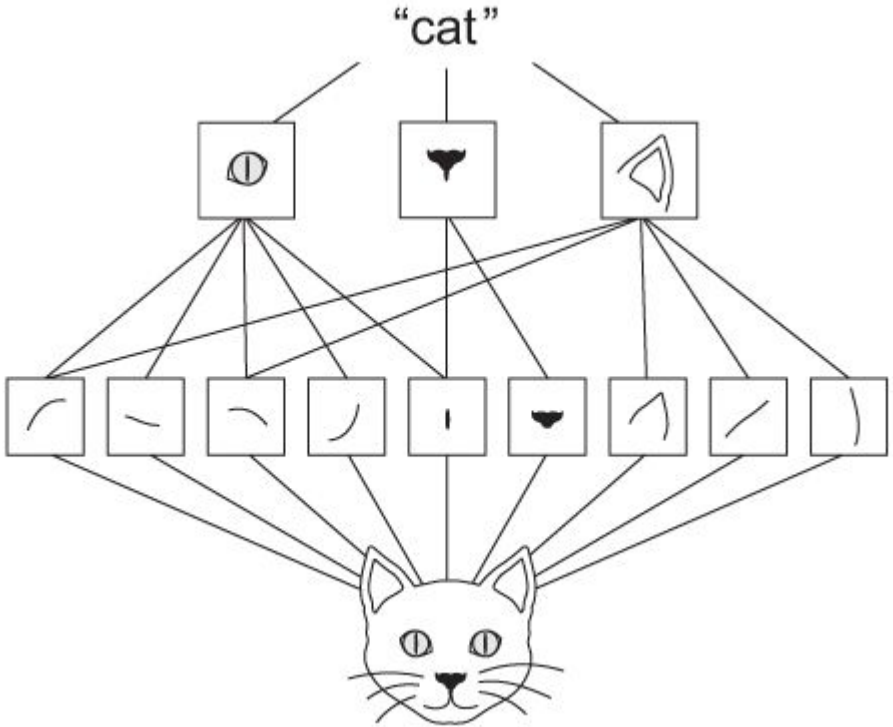
© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Transfer Learning - Feature Extraction



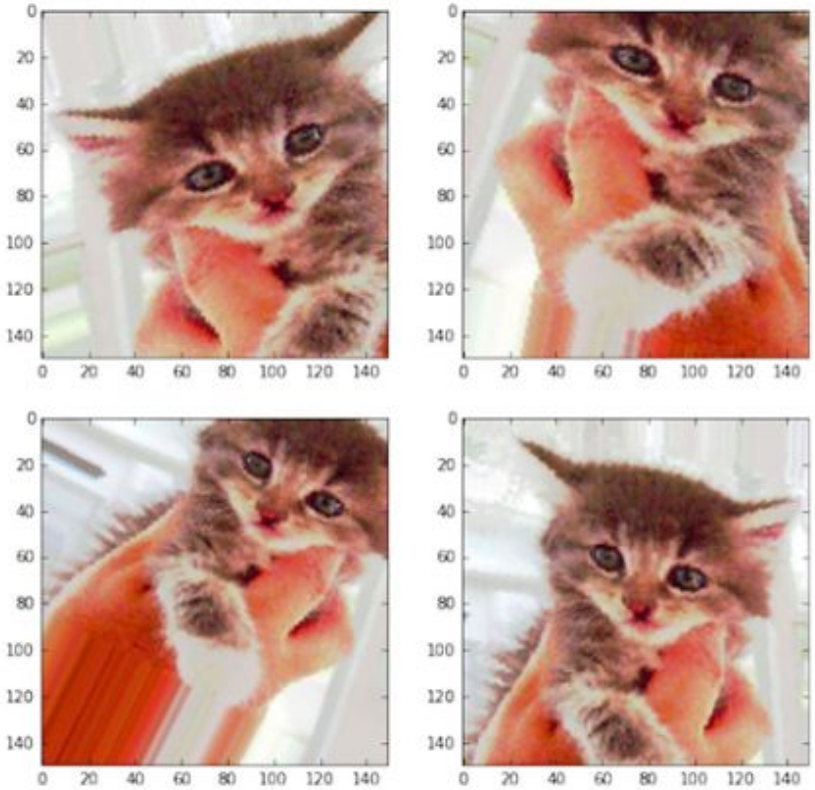
© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Learned Features



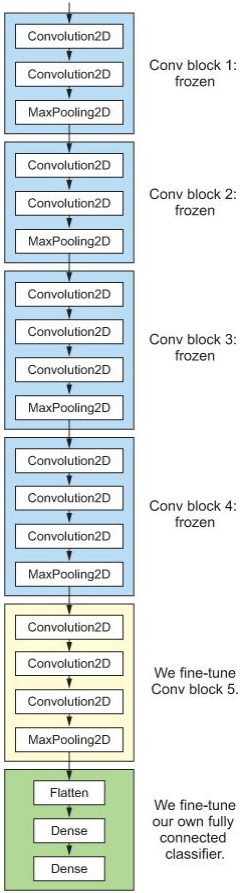
© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Data Augmentation



© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Transfer Learning - Fine Tuning



© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

Transfer Learning Demo!

© 2017 Google Inc. All rights reserved. Google and the Google logo are trademarks of Google Inc. All other company and product names may be trademarks of the respective companies with which they are associated.

A Structured Approach to Machine Learning

© 2017 Google Inc. All rights reserved.

Google Machine Learning Project Phases

We can assist clients in adopting a structured approach to developing ML solutions. The following a key phases that lead to a successful ML project

Phase 1: Define ML Use Case

Phase 2: Data Exploration

Phase 3: Select Algorithm

Phase 4: Data Pipeline & Feature Engineering

Phase 5: Build ML Model

Phase 6: Iterate to improve model performance

Phase 7: Present results, tell a story from data

Phase 8: Plan for Deployment

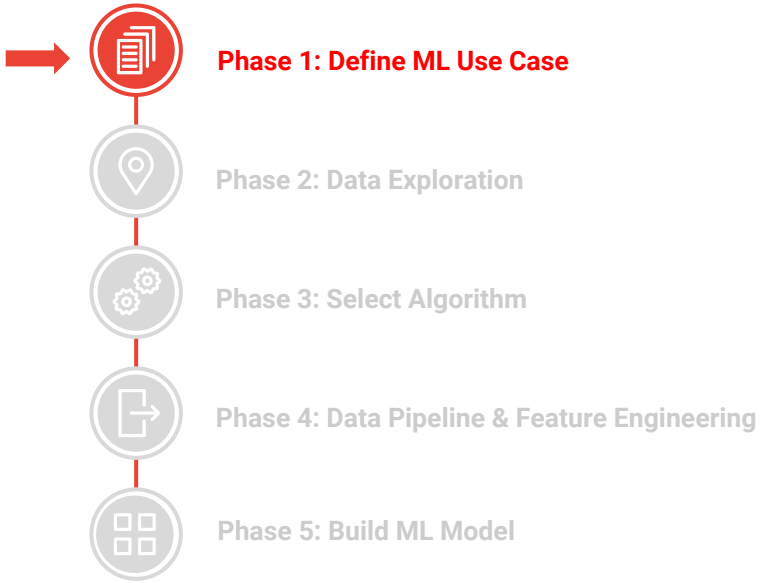
Phase 9: Deploy & Operationalization Model - Monitor

Google Cloud

© 2017 Google Inc. All rights reserved.

Machine Learning Project Phases

We can assist clients in adopting a structured approach to developing ML solutions. The following are key phases that lead to a successful ML project

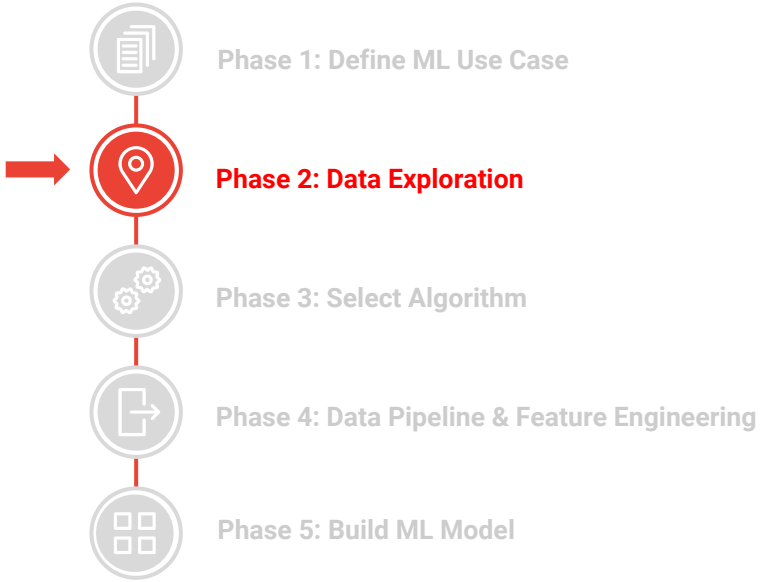


Define Use Case	Identify the business problem to be solved and potential business impact
Identify Hypothesis	Define potential patterns you expect to see in your data in order to test
Experiment	Define experiments you will need to test your hypothesis
Identify Assets	Identify needed data and subject matter experts to test the use case
Define Evaluation Metrics	Define measurable project success criteria

© 2017 Google Inc. All rights reserved.

Machine Learning Project Phases

We can assist clients in adopting a structured approach to developing ML solutions. The following are key phases that lead to a successful ML project



Phase 1: Assessment

- Describe Data
- Determine data quality
- Explore data through queries, visualization
- Identify correlations
- Define recommendation of use case, suggested next steps

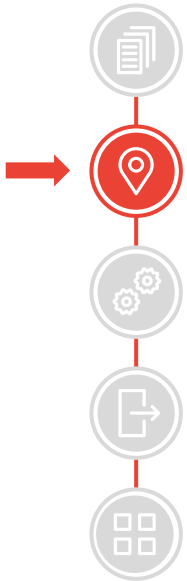
Phase 2: Deeper Analysis

- Identify patterns, outliers in data
- Checkpoint to validate hypothesis

© 2017 Google Inc. All rights reserved.

Machine Learning Project Phases

We can assist clients in adopting a structured approach to developing ML solutions. The following are key phases that lead to a successful ML project



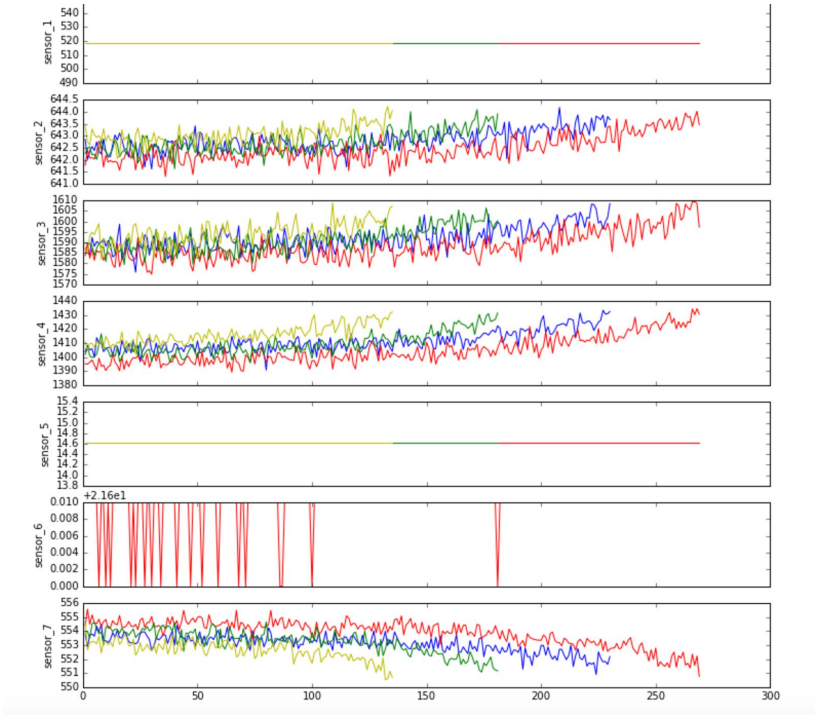
Phase 1: Define ML Use Case

Phase 2: Data Exploration

Phase 3: Select Algorithm

Phase 4: Data Pipeline & Feature Engineering

Phase 5: Build ML Model

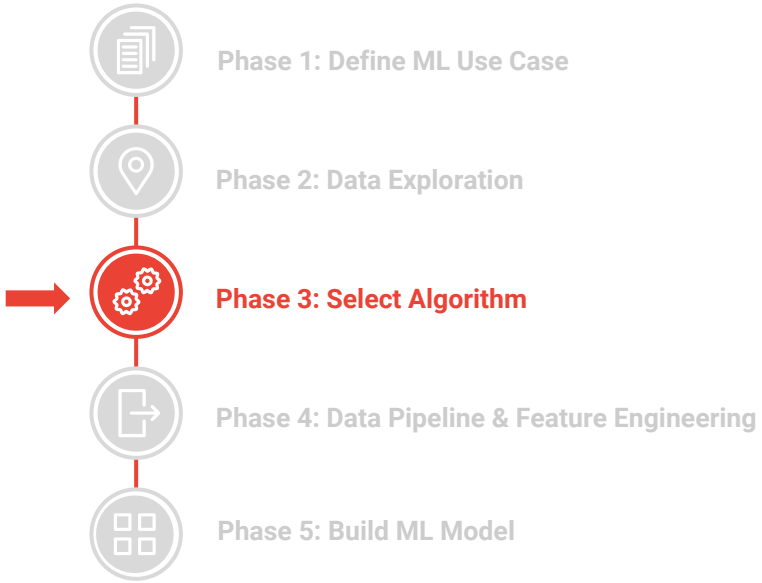


Google Cloud

© 2017 Google Inc. All rights reserved.

Machine Learning Project Phases

We can assist clients in adopting a structured approach to developing ML solutions. The following are key phases that lead to a successful ML project



1

Research
Leverage existing strategies and white papers to determine options for your use case

2

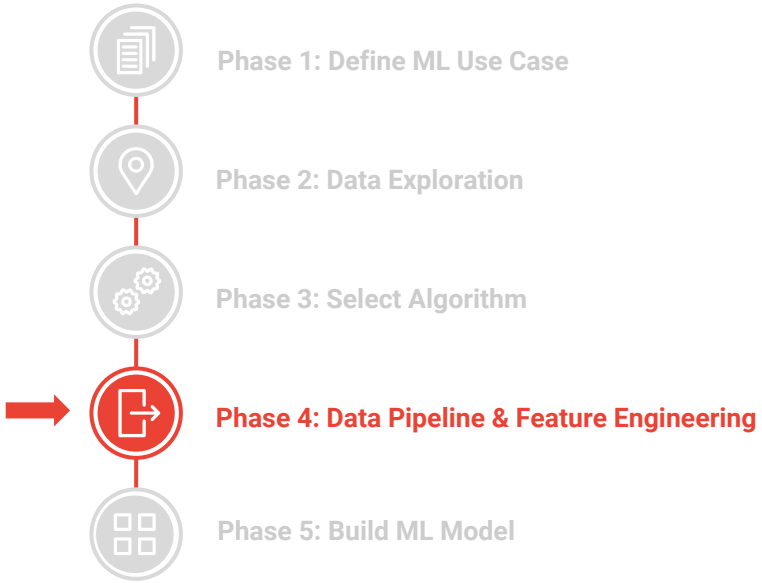
Select
Choose your algorithm based on initial hypothesis, and observed patterns in data

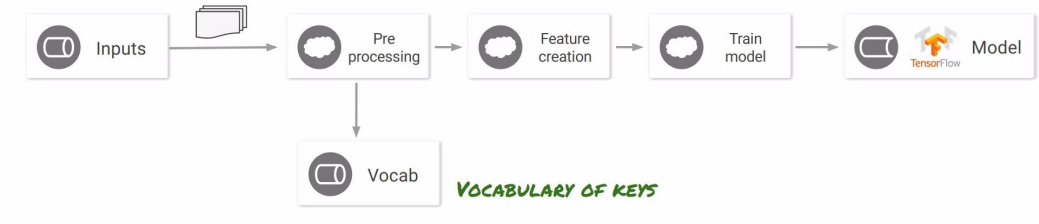
- Classification vs Regression
- Supervised vs Unsupervised
- Univariate or Multivariate
- Time Series
- DNN vs. Non DNN

© 2017 Google Inc. All rights reserved.

Machine Learning Project Phases

We can assist clients in adopting a structured approach to developing ML solutions. The following are key phases that lead to a successful ML project





What Makes a Good Feature

Represents raw data in a form conducive for ML

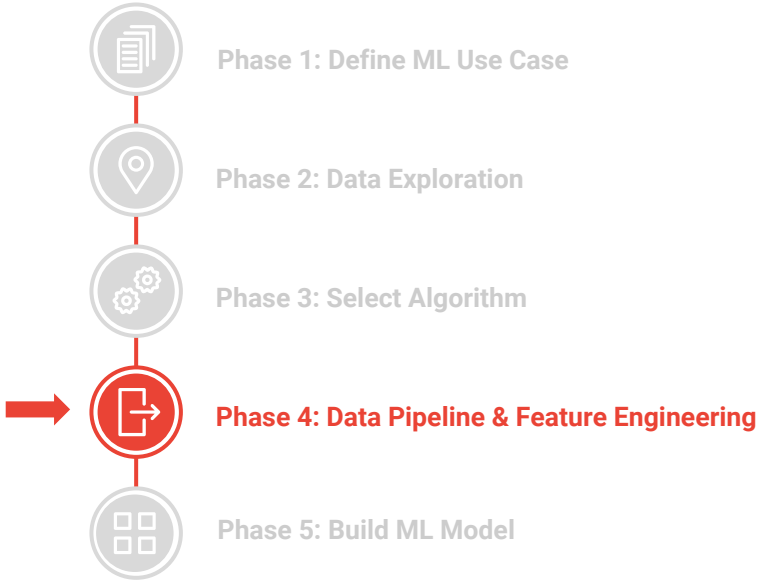
1. Should be related to the objective
2. Should be known at production-time
3. Has to be numeric with meaningful magnitude
4. Has enough examples
5. Brings human insight to problem

Google Cloud

© 2017 Google Inc. All rights reserved.

Machine Learning Project Phases

We can assist clients in adopting a structured approach to developing ML solutions. The following are key phases that lead to a successful ML project



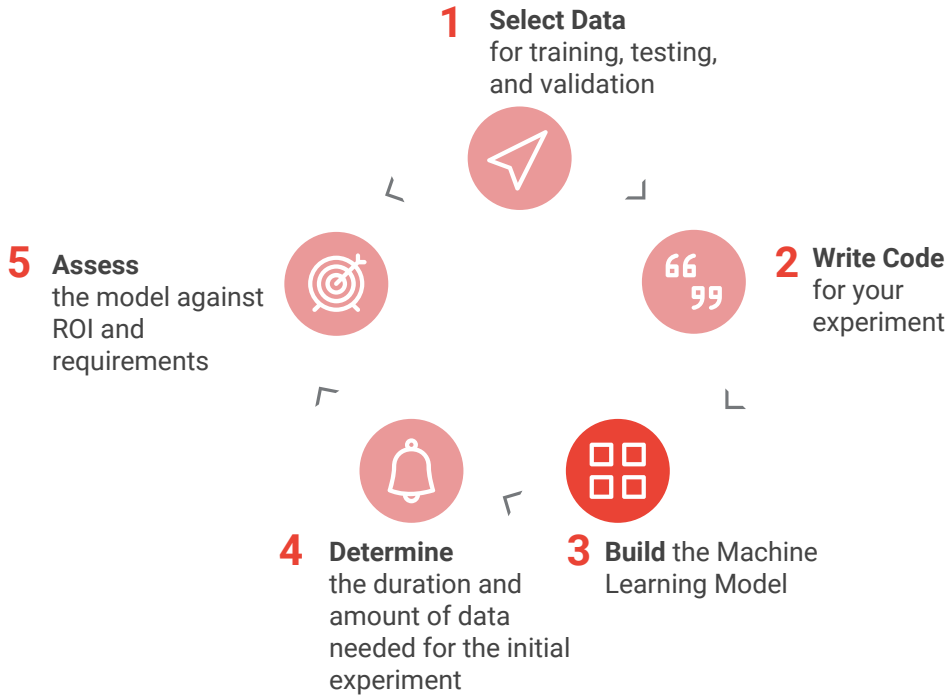
- Use domain knowledge to identify features
- Define new features: Distance between lat/long
- Remove redundant or duplicate features
- Reduce dimensionality as required
- Remove highly correlated features
- Check for class imbalance
- Deal with missing values and outliers

© 2017 Google Inc. All rights reserved.

Machine Learning Project Phases

We can assist clients in adopting a structured approach to developing ML solutions. The following are key phases that lead to a successful ML project

- Phase 1: Define ML Use Case
- Phase 2: Data Exploration
- Phase 3: Select Algorithm
- Phase 4: Data Pipeline & Feature Engineering
- **Phase 5: Build ML Model**



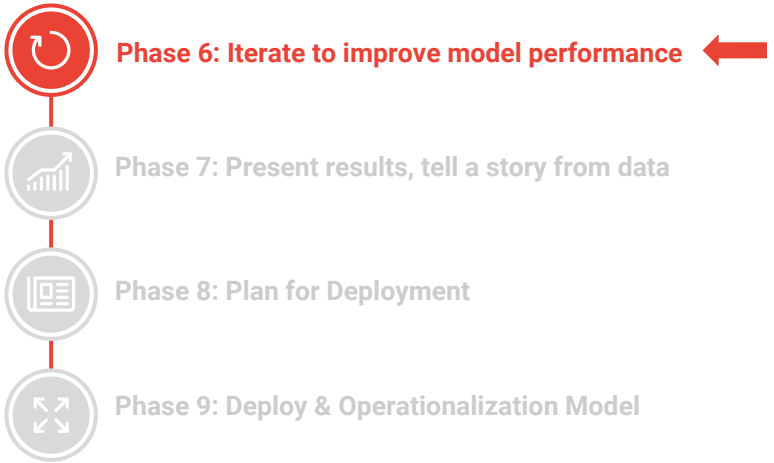
Google Cloud

© 2017 Google Inc. All rights reserved.

Machine Learning Project Phases

We can assist clients in adopting a structured approach to developing ML solutions. The following are key phases that lead to a successful ML project

1. Evaluate model result
2. Visualize model result
3. Determine correct actions (e.g. data cleanup, updates to features)
4. Iterate and Improve result

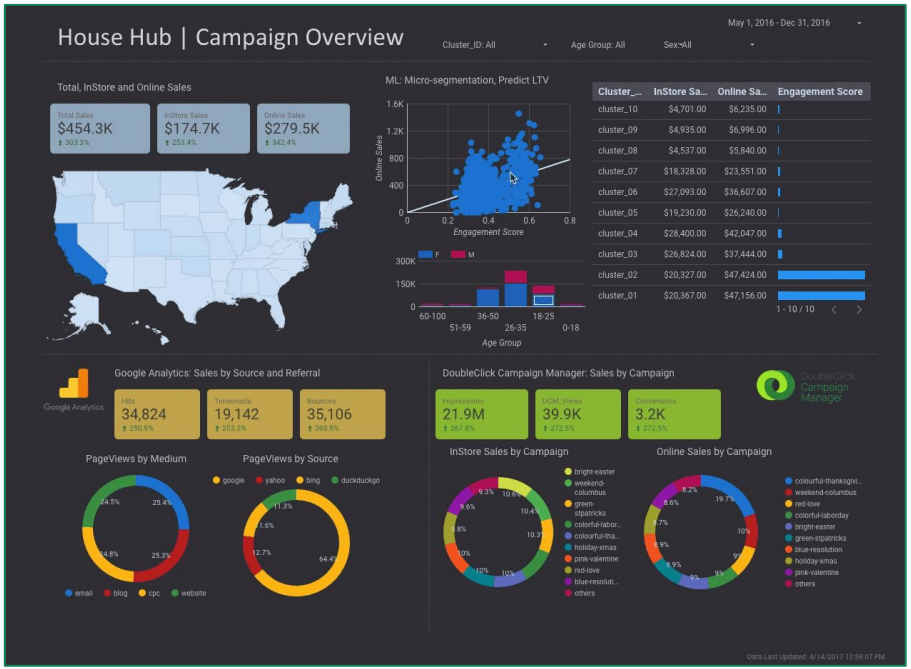


Google Cloud

© 2017 Google Inc. All rights reserved.

Machine Learning Project Phases

We can assist clients in adopting a structured approach to developing ML solutions. The following a key phases that lead to a successful ML project



Phase 6: Iterate to improve model performance

Phase 7: Present results, tell a story from data

Phase 8: Plan for Deployment

Phase 9: Deploy & Operationalization Model

© 2017 Google Inc. All rights reserved.

Machine Learning Project Phases

We can assist clients in adopting a structured approach to developing ML solutions. The following a key phases that lead to a successful ML project

01 Make prediction on production data and build a business case for operationalizing it

02 Prepare operationalization requirement for training, scoring

03 Prepare performance and scale requirement for production

04 Prepare architecture for model training, retraining

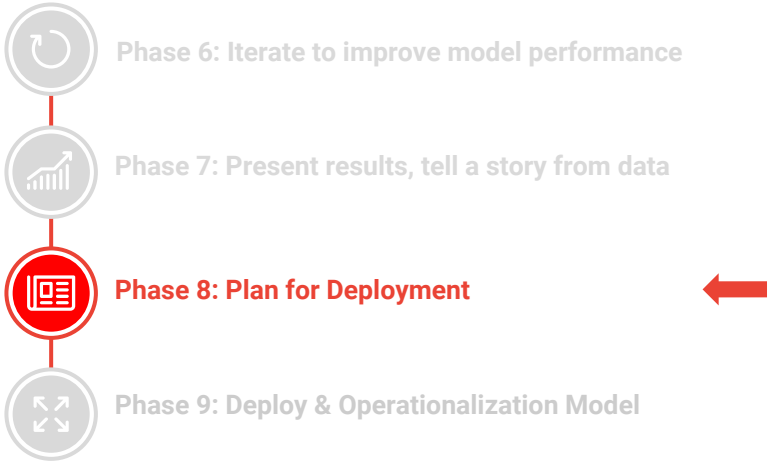
05 Plan to capture and save the model's performance metrics e.g. accuracy, precision, recall, AUC on a daily or weekly basis

06 Define architecture and integration for model

07 Prepare work breakdown structure

08 Develop proposed timelines for training/re-training model

09 Prepare plan for rollout and success criteria for increasing traffic.

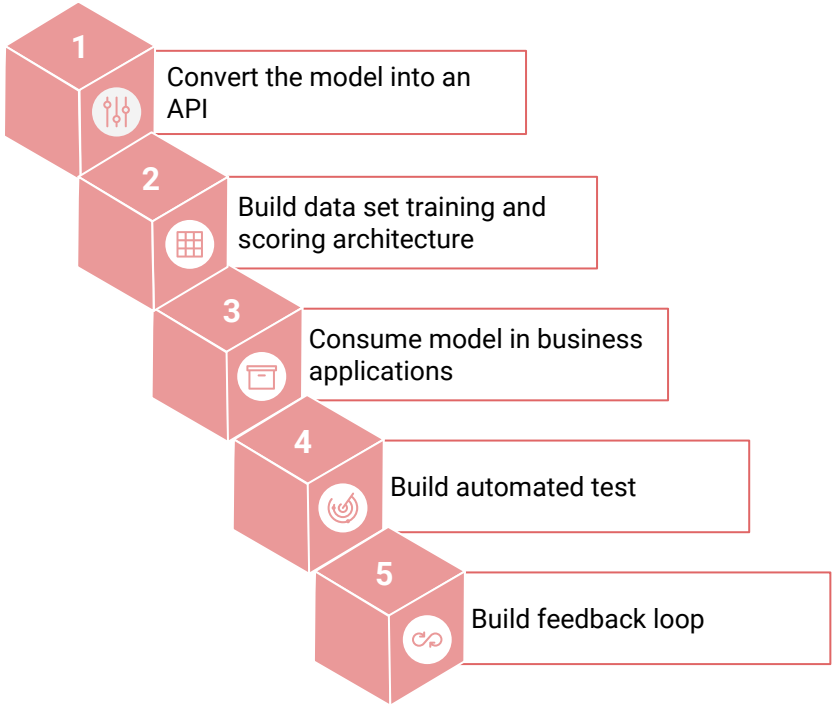


Google Cloud

© 2017 Google Inc. All rights reserved.

Machine Learning Project Phases

We can assist clients in adopting a structured approach to developing ML solutions. The following a key phases that lead to a successful ML project



Phase 6: Iterate to improve model performance

Phase 7: Present results, tell a story from data

Phase 8: Plan for Deployment

Phase 9: Deploy & Operationalization Model

Google Cloud

© 2017 Google Inc. All rights reserved.

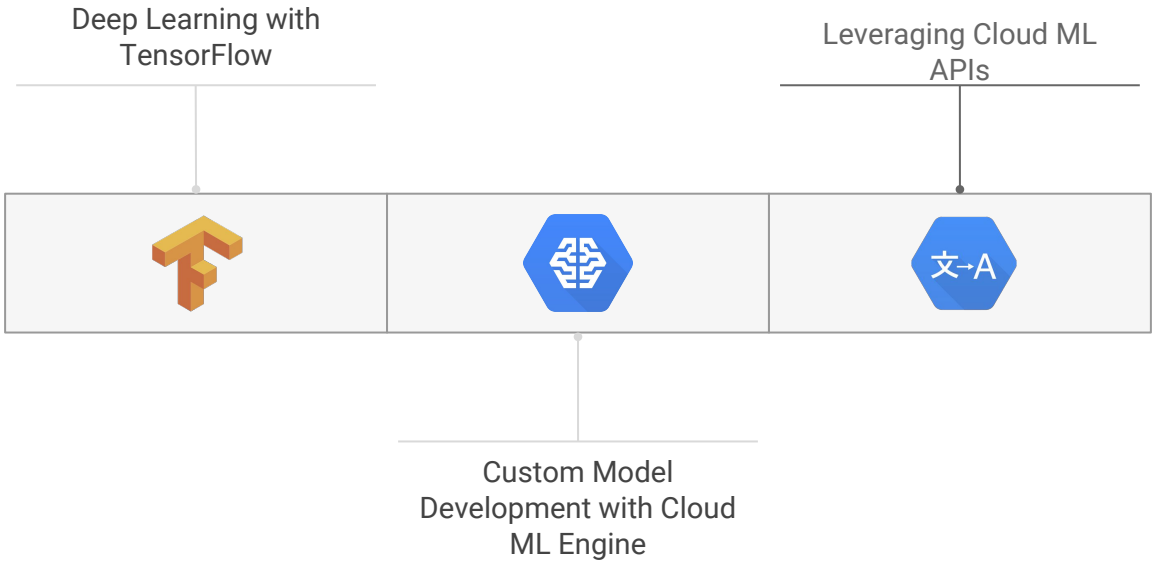
Section 1 Section 2 Section 3 Section 4

GCP Tools for ML

Proprietary + Confidential

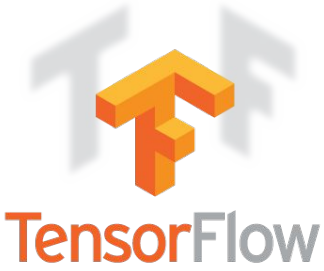
GCP Tools for ML

The following are services and tools for delivering ML on GCP



© 2017 Google Inc. All rights reserved.

The Machine Learning Spectrum on GCP



Opensource, not specific to GCP

CloudML

Scale, No-ops Infrastructure

ML APIs

Translate API

Vision API

Jobs API

Speech API

Language API

Video Intelligence API

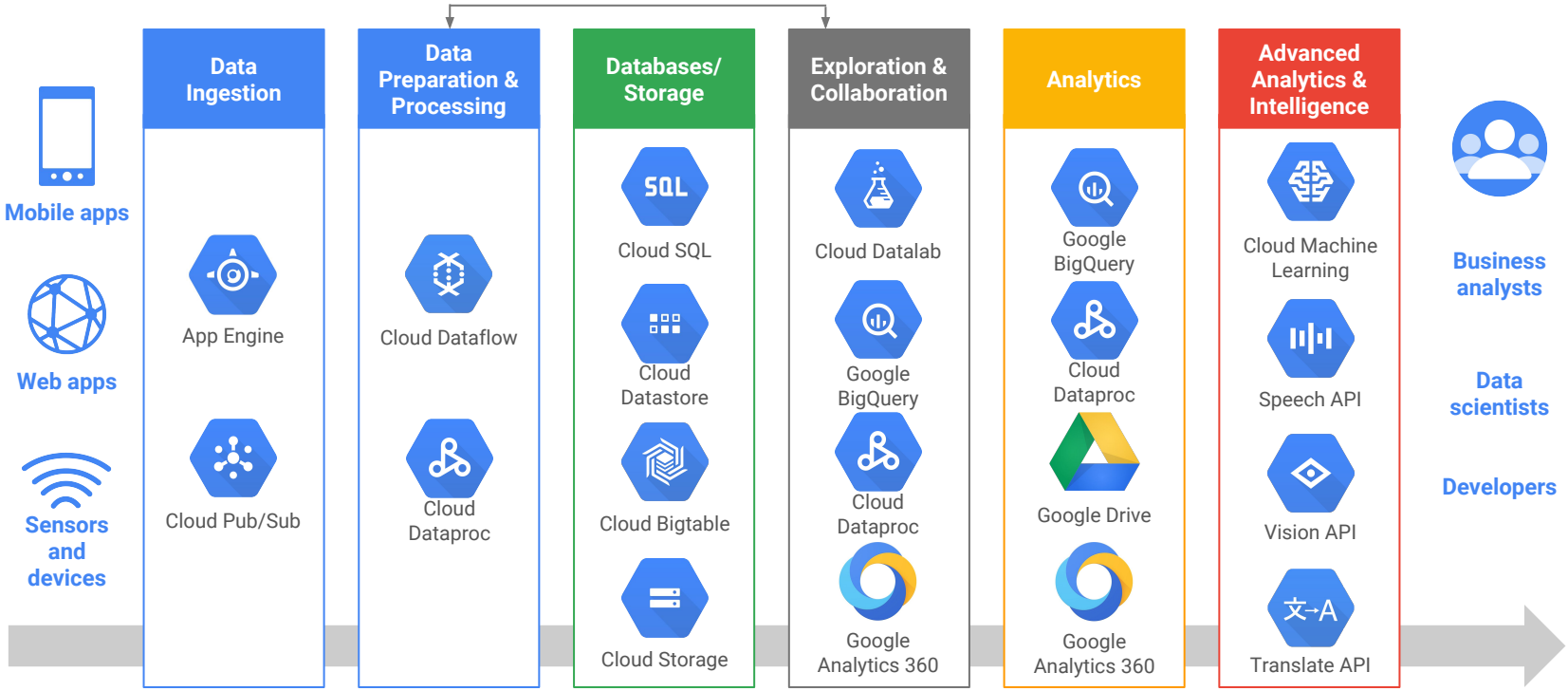
Use/extend OSS SDK | **Build custom models** | **Use pre-built models**

ML researcher | **Data Scientist** | **App Developer**

Google Cloud

© 2017 Google Inc. All rights reserved.

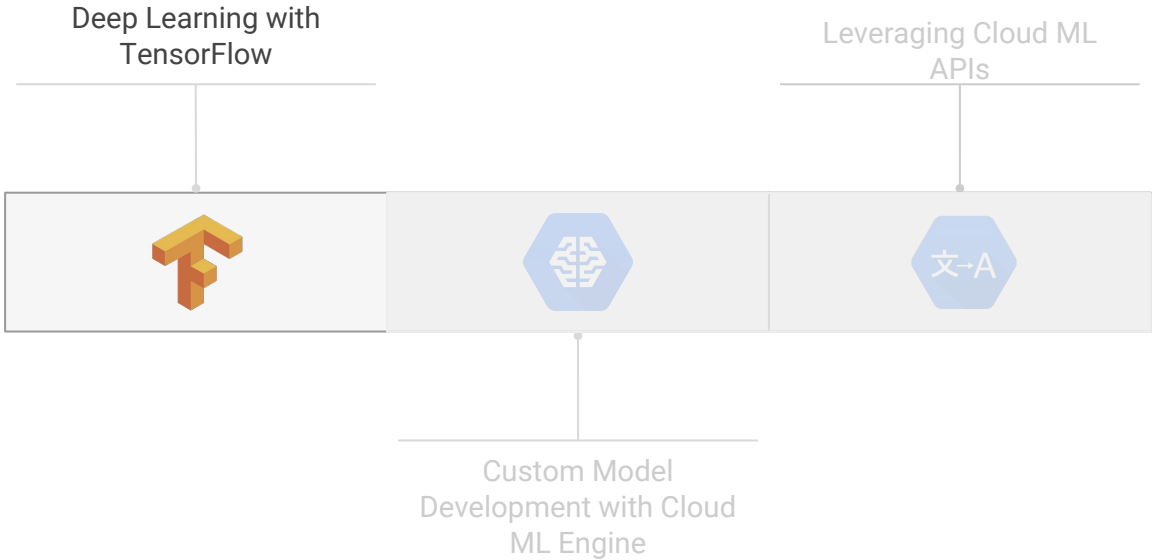
Transform data into actions



© 2017 Google Inc. All rights reserved.

GCP Tools for ML

The following are services and tools for delivering ML on GCP



© 2017 Google Inc. All rights reserved.

Deep Learning with TensorFlow

© 2017 Google Inc. All rights reserved.

Built on Open Source

Created by Google Brain team

Most popular ML project on Github

- Over 480 contributors
- 10,000 commits in 12 months

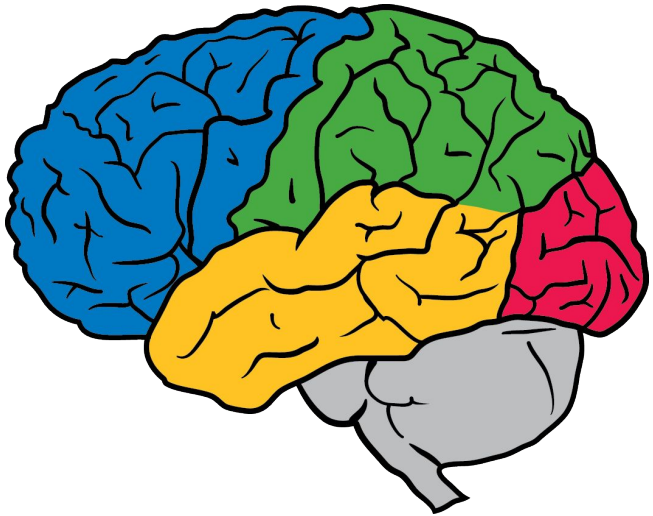
Multiple deployment options:

- Mobile, Desktop, Server, Cloud
- CPU, GPU

What is TensorFlow?

A new system for distributed, parallel machine learning:

- Based on general-purpose dataflow graphs
- Great for constructing neural nets
- Leverages GPUs



Training with TensorFlow

Two phases: graph construction and graph execution.

Graph Construction

- Define a skeleton for your model, but don't run any computations yet!

Graph Execution

- Feed training set to graph in batches and compute optimal weights!

This approach is known as *lazy evaluation* and gives performance gains by avoiding context switching!

© 2017 Google Inc. All rights reserved.

Graph Construction

- TensorFlow provides a high-level APIs (tf.learn, tf.Estimator) to do the heavy lifting - you'll never have to implement a neural net from scratch!

- Specify training set (csv, json) and perform any necessary preprocessing (TensorFlow expects input data in a specific json format)

- Set hyperparameters
  - Training Rate
  - Batch Size
  - Number of Hidden Layers
  - Sizes of Hidden Layers

© 2017 Google Inc. All rights reserved.

Graph Evaluation

- Run training job in the cloud with CloudML!
- Leverages GPUs and distributed, parallel infrastructure for massive performance gains.
- Outputs (e.g. your trained model) are written to GCS. Logs available in Stack Driver.

© 2017 Google Inc. All rights reserved.

Prediction

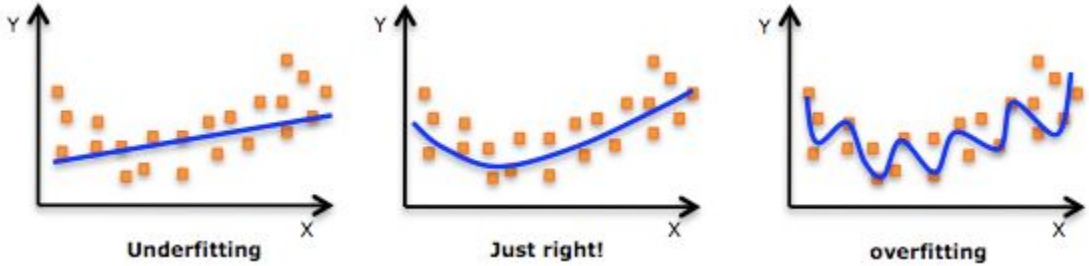
- Cloud ML automatically hosts deployed models, and manages serving instances to handle prediction requests.

- Provides an HTTP prediction API

© 2017 Google Inc. All rights reserved.

Evaluating Your Model

- Training Set, Cross Validation Set, and Test Set



- Precision
  - When the boy cried wolf, how often was he right?

- Recall
  - When there was a wolf, how often did the boy cry wolf?

© 2017 Google Inc. All rights reserved.

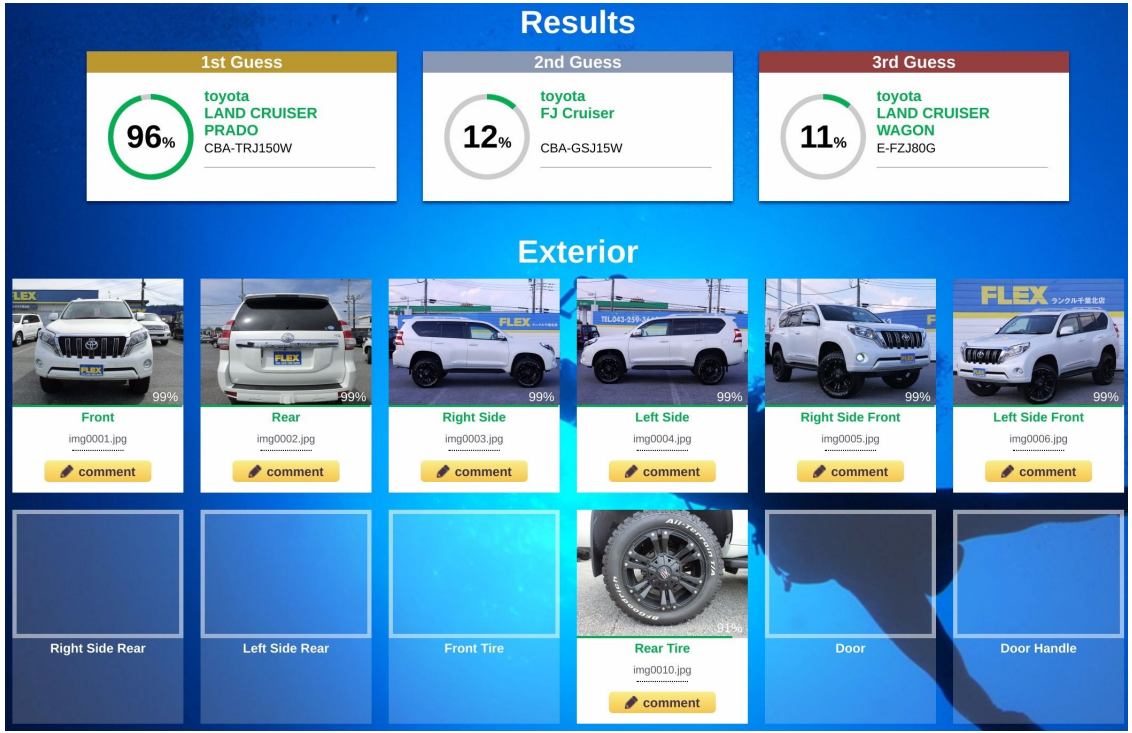
Aucnet: Image Classification with TensorFlow

Aucnet developed a PoC of UI for used-car dealers that classifies cars by model and by physical parts by simply uploading pictures

Developed on TensorFlow using multiple deep learning models and 5000 images as its data corpus

TensorFlow enabled almost 100% accuracy in classifying cars by make and model

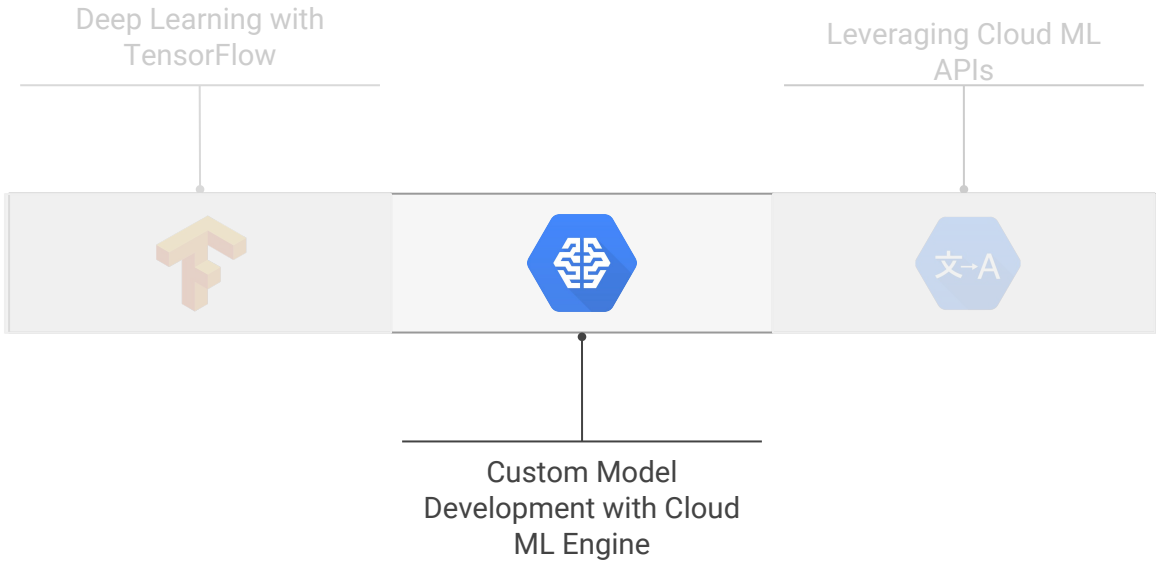
Demo



© 2017 Google Inc. All rights reserved.

GCP Tools for ML

The following are services and tools for delivering ML on GCP



© 2017 Google Inc. All rights reserved.

Custom Model Development with Cloud Machine Learning Engine

© 2017 Google Inc. All rights reserved.

Cloud Machine Learning Engine

- Fully managed service
- Train using a custom TensorFlow graph for any ML use cases with CPUs/GPUs
- Training at scale to shorten dev cycle
- Automatically maximize predictive accuracy with HyperTune
- High throughput batch predictions
- Low latency online predictions (Beta)
- Integrated Datalab experience

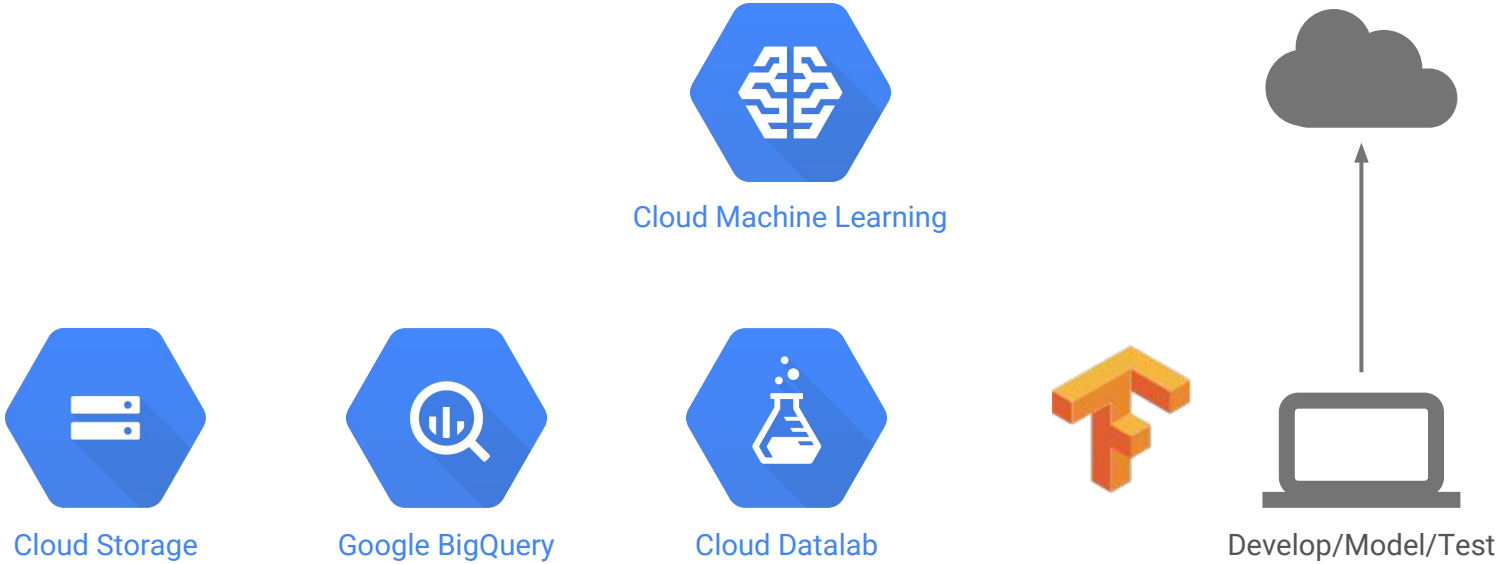
© 2017 Google Inc. All rights reserved.

Cloud Datalab

- Interactively explore data
- Define features with rich visualization support
- Launch training and evaluation
- ML lifecycle support
- Combine code, results, visualizations & documentation in notebook format
- Share results with your team
- Pick from a rich set of tutorials & samples to learn and get started with your project

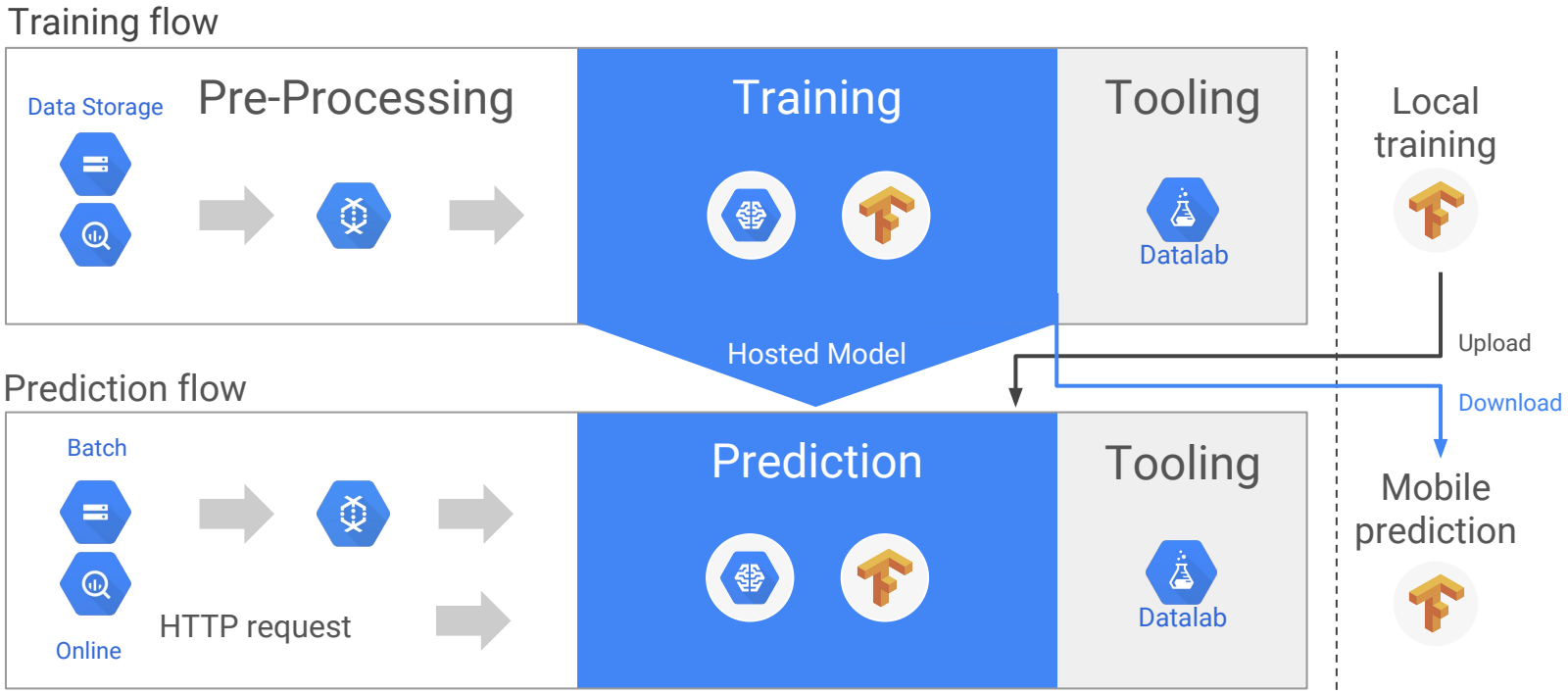
© 2017 Google Inc. All rights reserved.

Training the Machine Learning Model



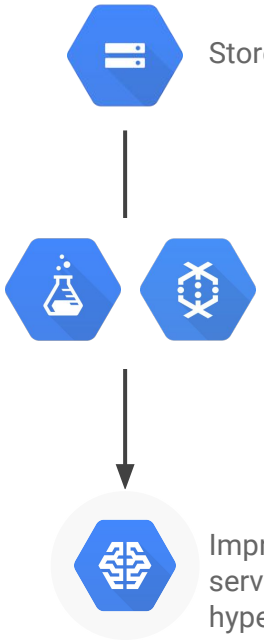
© 2017 Google Inc. All rights reserved.

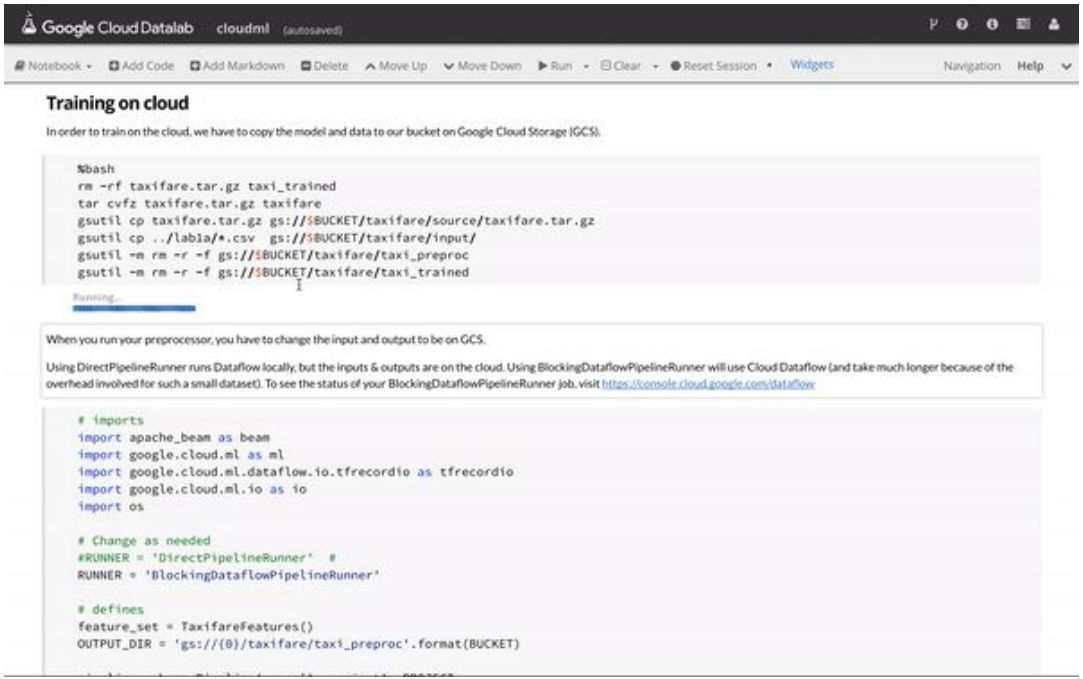
Operationalizing the Model



© 2017 Google Inc. All rights reserved.

Scale it out to GCP using serverless technology

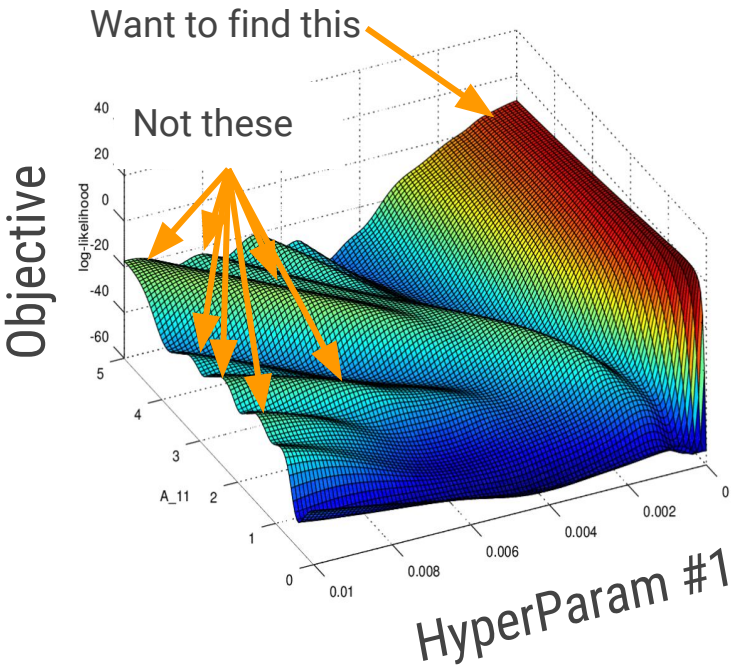




© 2017 Google Inc. All rights reserved.

Automatically tune your model with HyperTune

- Automatic hyperparameter tuning service
- Build better performing models faster and save many hours of manual tuning
- Google-developed search algorithm efficiently finds better hyperparameters for your model/dataset



© 2017 Google Inc. All rights reserved.

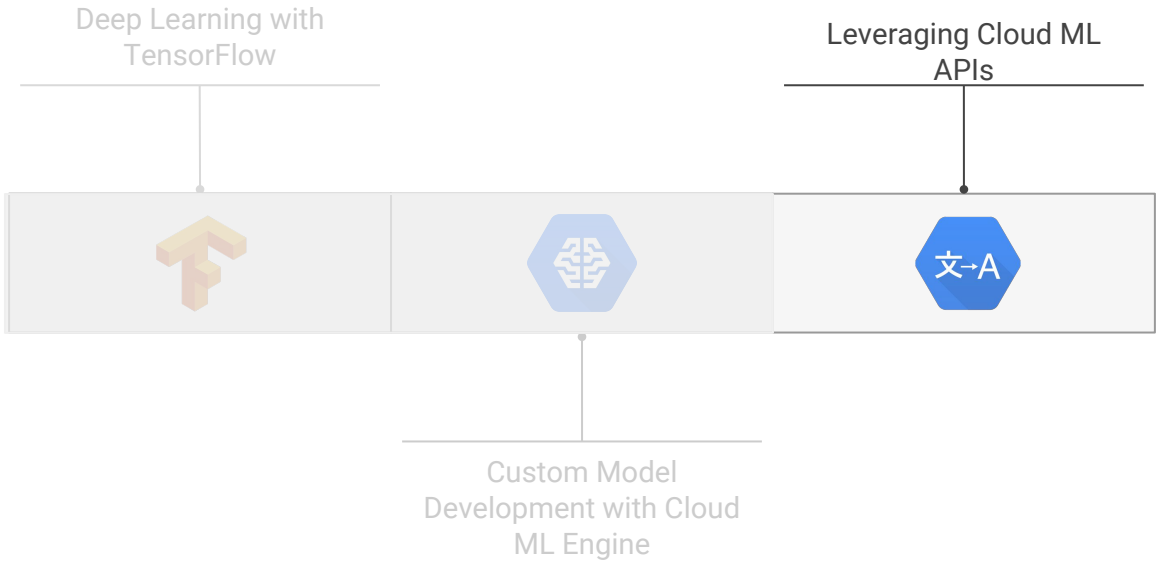
Integrated with GCP products

- Access data that is stored in GCS or BigQuery
- Save trained models to GCS
- Preprocess largest datasets (TB) using Dataflow
- Orchestrate ML workflow as a Dataflow pipeline
- Analyze data and interactively develop ML models in Datalab

© 2017 Google Inc. All rights reserved.

GCP Tools for ML

The following are services and tools for delivering ML on GCP



Google Cloud

© 2017 Google Inc. All rights reserved.

Leveraging Cloud ML APIs

© 2017 Google Inc. All rights reserved.

Google offers a full spectrum of offerings

Customizable, for Data Scientists

Easy-to-Use, for non-ML engineers

TensorFlow

Cloud Machine Learning

- Cloud Machine Learning Engine
- Cloud Dataproc (w/Spark ML)

ML API

- Translate API
- Vision API
- Speech API
- Natural Language API
- Jobs API
- Video Intelligence API

© 2017 Google Inc. All rights reserved.

Cloud ML APIs

Translate API

Vision API

Jobs API

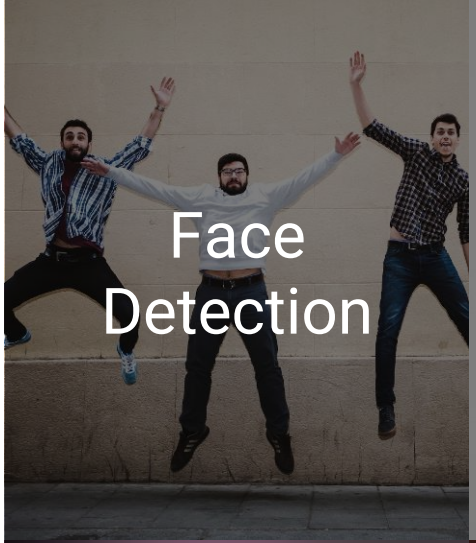
Speech API

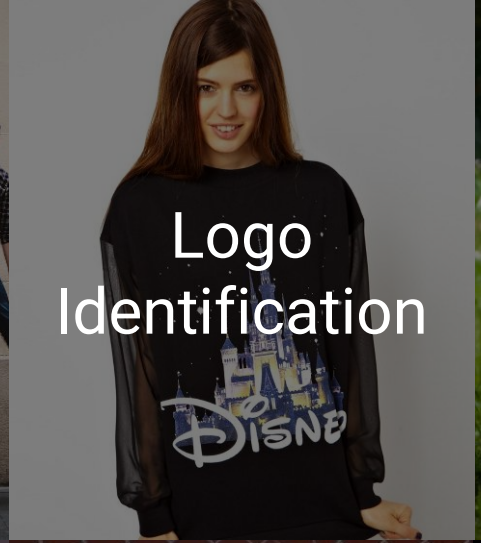
Language API

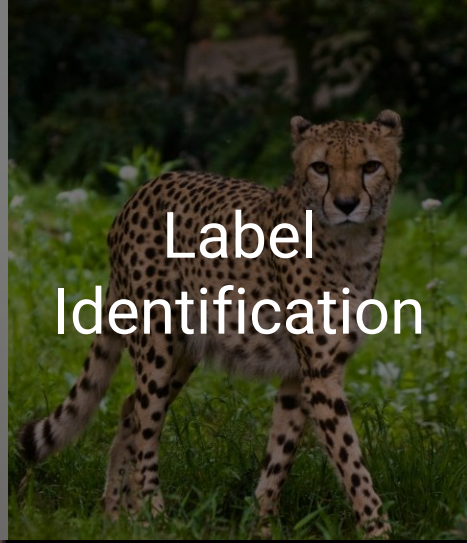
Video Intelligence API

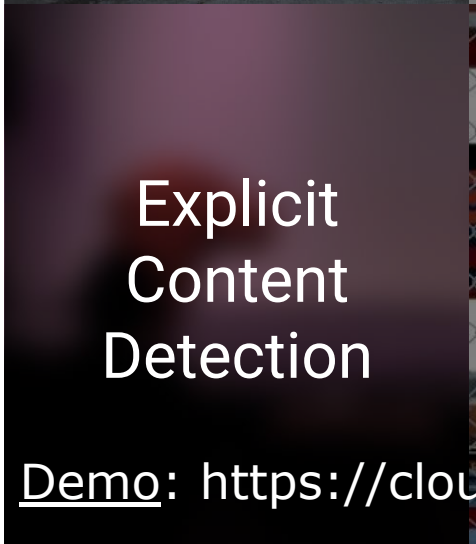
© 2017 Google Inc. All rights reserved.

Cloud Vision









Demo: https://cloud.google.com/vision/

Cloud Speech

Recognizes over 80 languages and variants

Supports real-time conversion

Inappropriate content filtering

Highly accurate, even in noisy environments

Access from any device

“Hello”

“Bonjour”

“Hola”

“Здравствуйте”

“안녕하세요”

“こんにちは”

“नमस्ते”

Demo:cloud.google.com/speech/

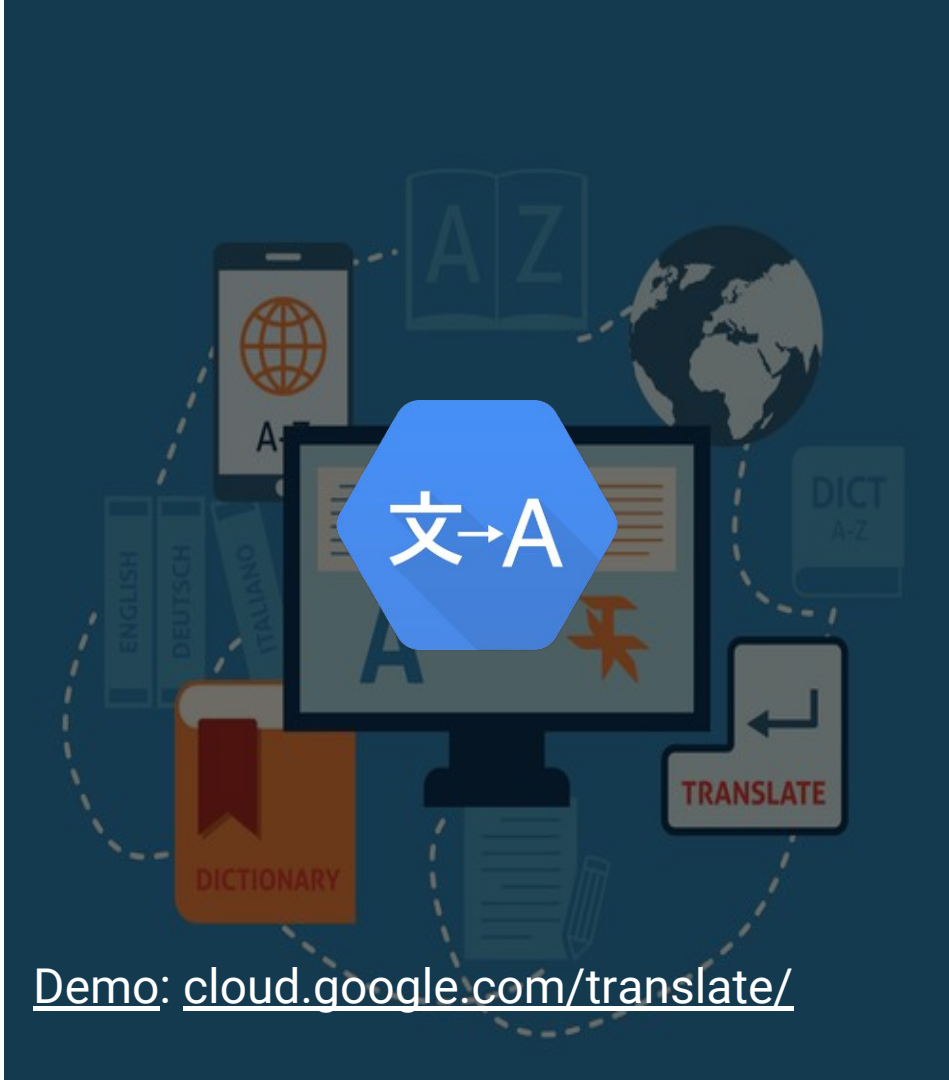
Cloud Translate

Supports more than 100 languages, from Afrikaans to Zulu

Detect language with high accuracy

Highly responsive to enable dynamic translation

Constantly updated to improve translations



Demo: cloud.google.com/translate/

Natural Language

Understand **structure** and **meaning**

Extract people, places, events, and more

Combine with Speech, Vision, and Translate for multimedia and multilingual understanding

Demo:
cloud.google.com/natural-language/

Video Intelligence

Search and discover video content by video, scene, or frame level

Extract information from content buried in noise

Detect entities and scene changes within the video

Improves over time as new concepts are introduced

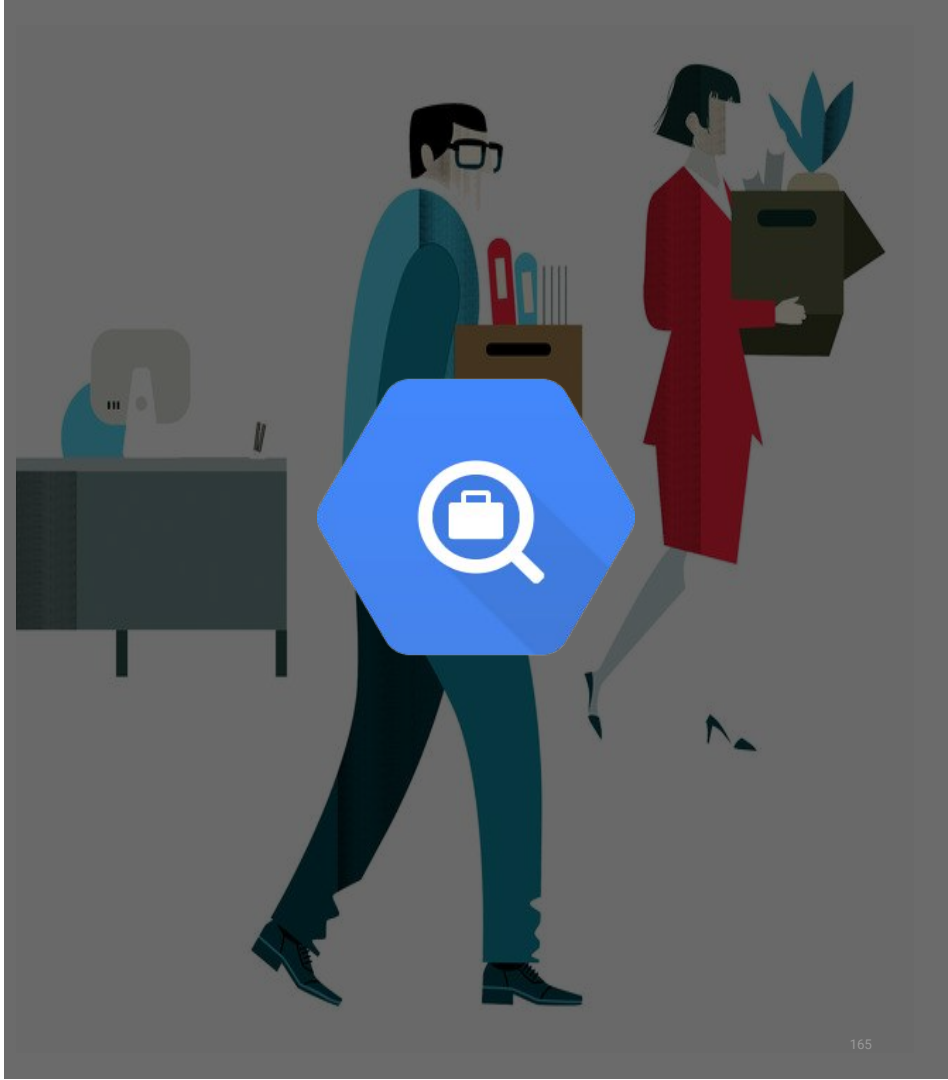
Demo: cloud.google.com/video-intelligence/

Cloud Jobs

Power job sites with Google’s search and machine learning capabilities

Delivers relevant search results by understanding the relationship between job content and other signals

Promotes jobs that fit talent’s location and commute preferences



Demo: End to End Demo with ML APIs

Thank You!

Proprietary + Confidential