



Cloud AutoML



Torry Yang
Developer Programs Engineer
@torryyang

Agenda

1

Cloud AutoML?

2

Vision

3

BTS

4

Natural
Language

5

Translation



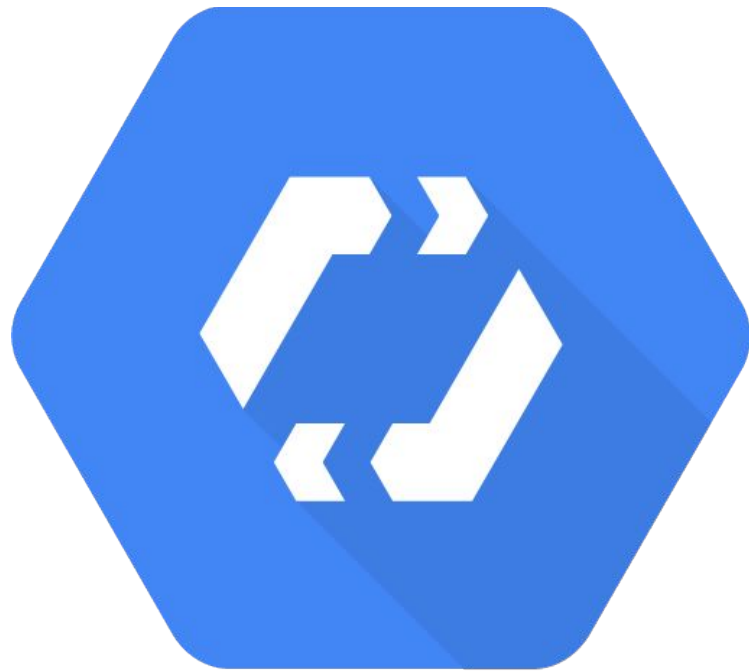
Google Cloud

Cloud AutoML

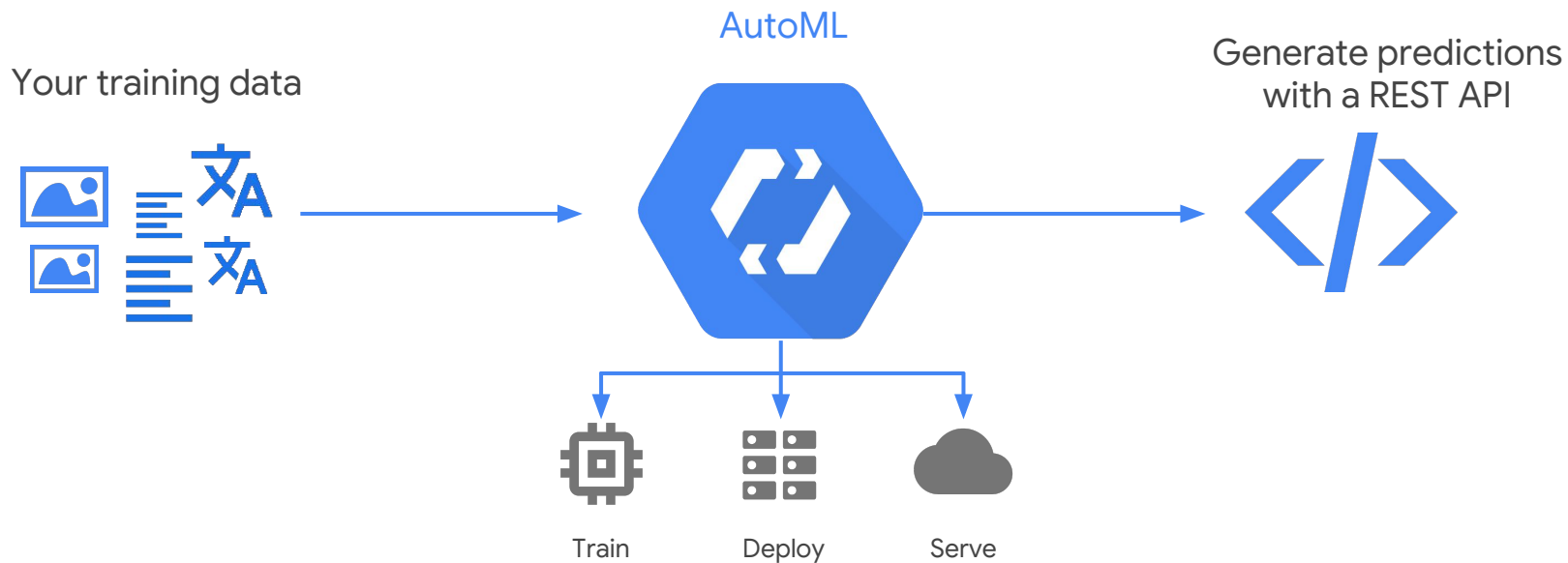


Cloud AutoML

Custom machine learning models
without model code



What does it do?



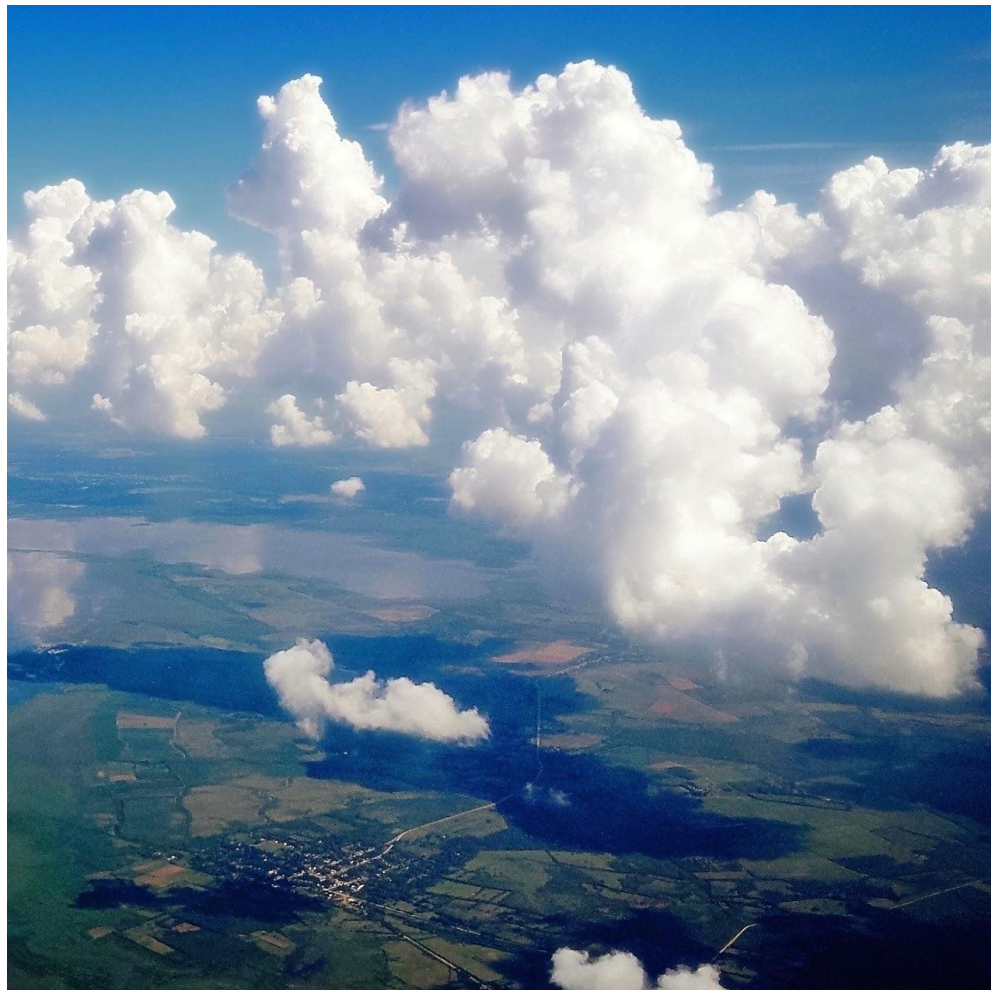
Vision





Let's say I'm a
meteorologist

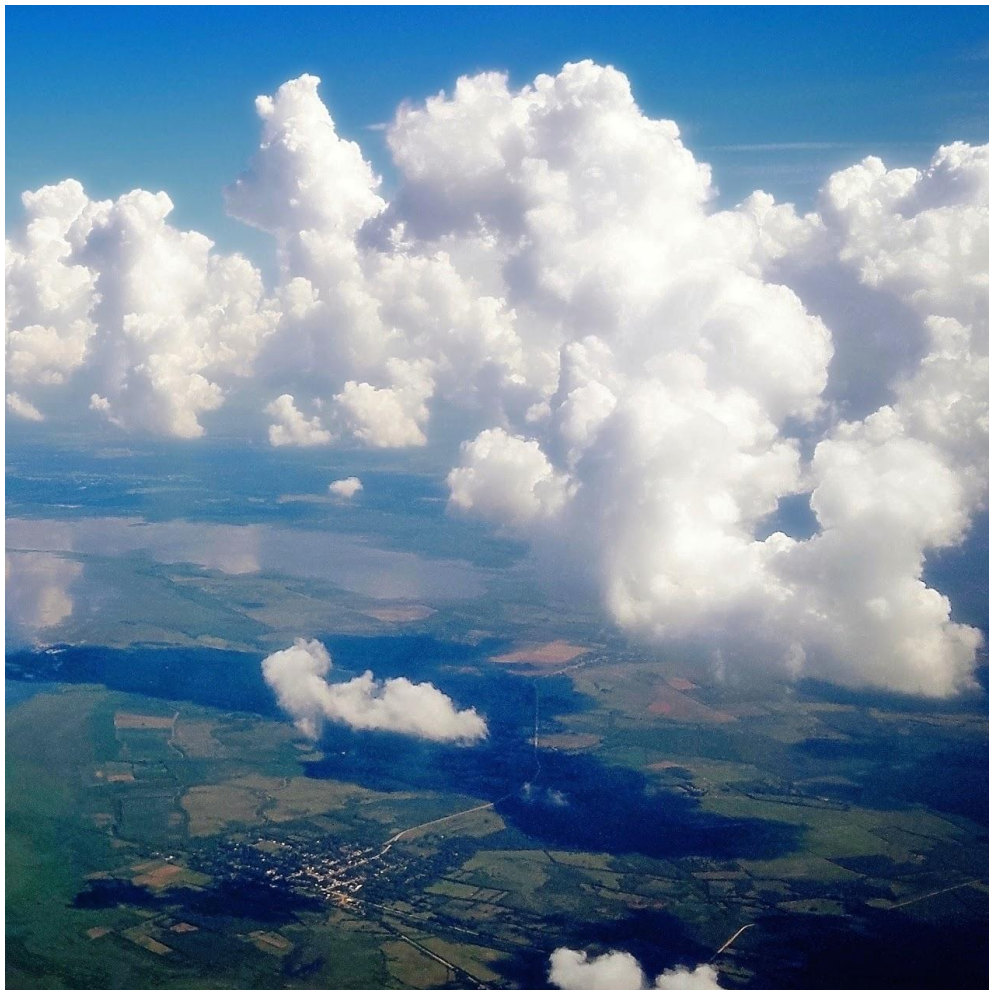
Google Cloud





I want to predict
weather trends
and flight plans

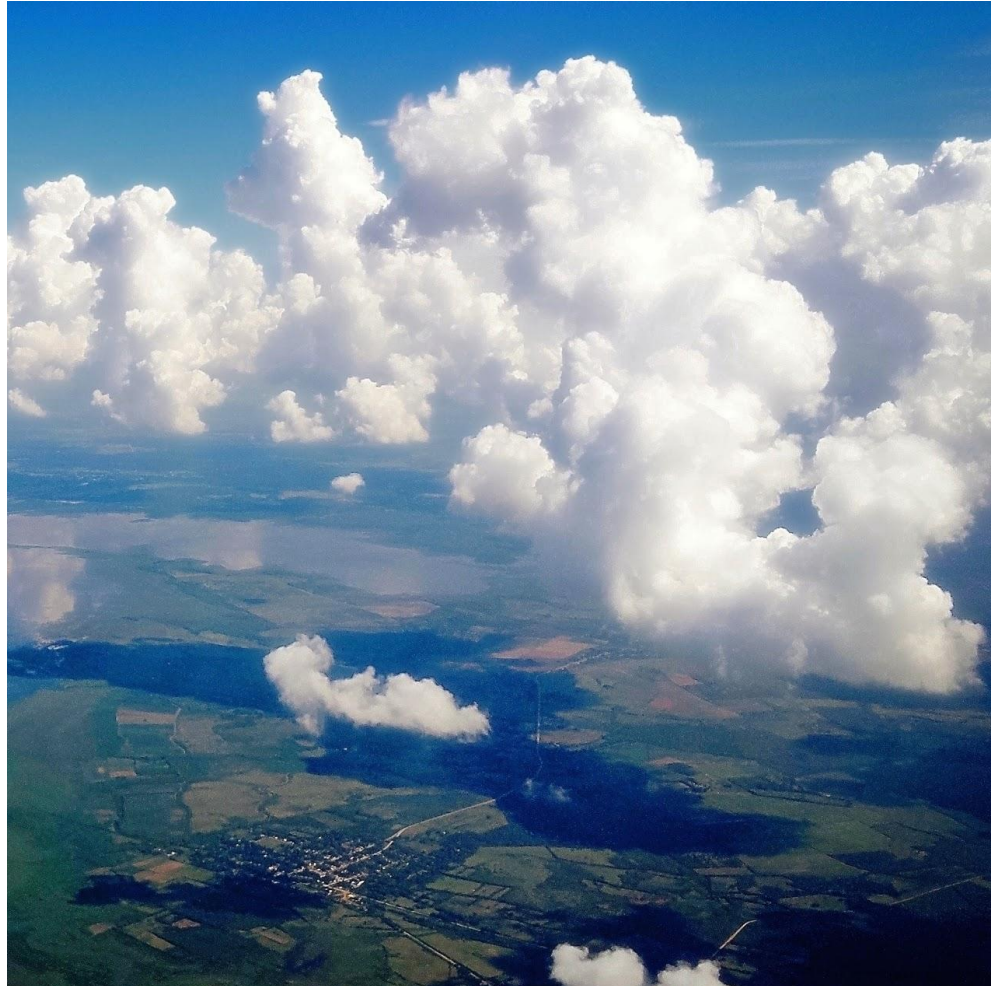
Google Cloud





Can we use the
cloud to analyze
clouds?

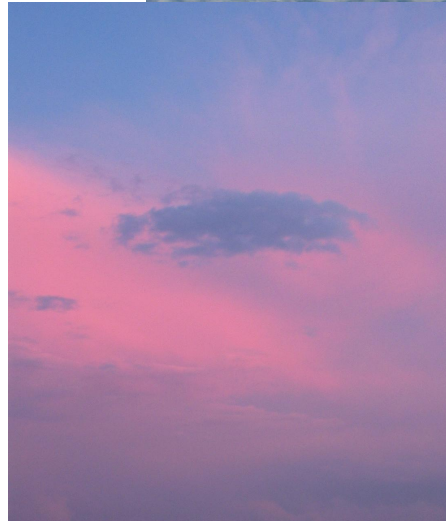
Google Cloud





There are **many**
types of clouds

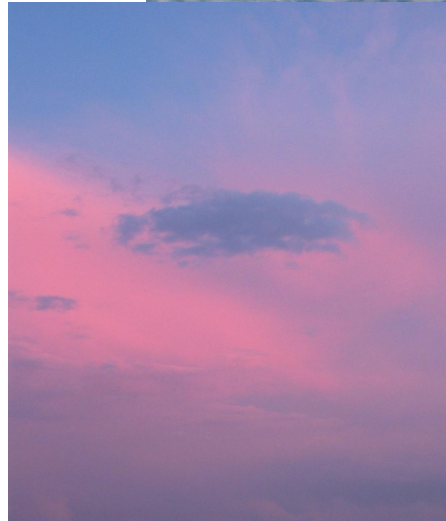
Google Cloud



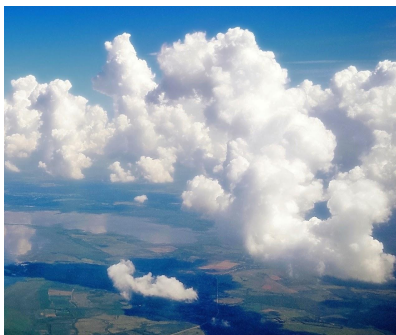


They all indicate
different weather
patterns

Google Cloud



Let's try the Vision API



Live demo: AutoML Vision

Create dataset

```
from google.cloud import automl_v1beta1 as automl
client = automl.AutoMLClient()
```

```
project_location = client.location_path(project_id, compute_region)
dataset_metadata = {"classification_type": MULTICLASS_or_MULTILABLE}
my_dataset = {
    "display_name": dataset_name,
    "image_classification_dataset_metadata": dataset_metadata
}
```

```
dataset = client.create_dataset(project_location, my_dataset)
```

Create dataset

```
from google.cloud import automl_v1beta1 as automl
client = automl.AutoMLClient()
```

```
project_location = client.location_path(project_id, compute_region)
dataset_metadata = {"classification_type": MULTICLASS_or_MULTILABLE}
my_dataset = {
    "display_name": dataset_name,
    "image_classification_dataset_metadata": dataset_metadata
}
```

```
dataset = client.create_dataset(project_location, my_dataset)
```

Create dataset

```
from google.cloud import automl_v1beta1 as automl
client = automl.AutoMLClient()
```

```
project_location = client.location_path(project_id, compute_region)
dataset_metadata = {"classification_type": MULTICLASS_or_MULTILABLE}
my_dataset = {
    "display_name": dataset_name,
    "image_classification_dataset_metadata": dataset_metadata
}
```

```
dataset = client.create_dataset(project_location, my_dataset)
```

Create dataset

```
from google.cloud import automl_v1beta1 as automl
client = automl.AutoMLClient()

project_location = client.location_path(project_id, compute_region)
dataset_metadata = {"classification_type": MULTICLASS_or_MULTILABLE}
my_dataset = {
    "display_name": dataset_name,
    "image_classification_dataset_metadata": dataset_metadata
}

dataset = client.create_dataset(project_location, my_dataset)
```

Create dataset

```
from google.cloud import automl_v1beta1 as automl
client = automl.AutoMLClient()
```

```
project_location = client.location_path(project_id, compute_region)
dataset_metadata = {"classification_type": MULTICLASS_or_MULTILABLE}
my_dataset = {
    "display_name": dataset_name,
    "image_classification_dataset_metadata": dataset_metadata
}
```

```
dataset = client.create_dataset(project_location, my_dataset)
```

Import data

```
from google.cloud import automl_v1beta1 as automl
client = automl.AutoMLClient()
```

```
dataset_full_id = client.dataset_path(
    project_id, compute_region, dataset_id
)
```

```
input_config = {"gcs_source": {"input_uris": input_uris}}
```

```
response = client.import_data(dataset_full_id, input_config)
```

Import data

```
from google.cloud import automl_v1beta1 as automl
client = automl.AutoMLClient()
```

```
dataset_full_id = client.dataset_path(
    project_id, compute_region, dataset_id
)
```

```
input_config = {"gcs_source": {"input_uris": input_uris}}
```

```
response = client.import_data(dataset_full_id, input_config)
```

Import data

```
from google.cloud import automl_v1beta1 as automl
client = automl.AutoMLClient()
```

```
dataset_full_id = client.dataset_path(
    project_id, compute_region, dataset_id
)
```

```
input_config = {"gcs_source": {"input_uris": input_uris}}
```

```
response = client.import_data(dataset_full_id, input_config)
```

Import data

```
from google.cloud import automl_v1beta1 as automl
client = automl.AutoMLClient()
```

```
dataset_full_id = client.dataset_path(
    project_id, compute_region, dataset_id
)
```

```
input_config = {"gcs_source": {"input_uris": input_uris}}
```

```
response = client.import_data(dataset_full_id, input_config)
```

Create model

```
from google.cloud import automl_v1beta1 as automl
client = automl.AutoMLClient()
```

```
project_location = client.location_path(project_id, compute_region)
```

```
my_model = {
    "display_name": model_name,
    "dataset_id": dataset_id,
    "image_classification_model_metadata": {"train_budget": train_budget}
}
```

```
response = client.create_model(project_location, my_model)
```

Create model

```
from google.cloud import automl_v1beta1 as automl
client = automl.AutoMLClient()
```

```
project_location = client.location_path(project_id, compute_region)
```

```
my_model = {
    "display_name": model_name,
    "dataset_id": dataset_id,
    "image_classification_model_metadata": {"train_budget": train_budget}
}
```

```
response = client.create_model(project_location, my_model)
```

Create model

```
from google.cloud import automl_v1beta1 as automl
client = automl.AutoMLClient()

project_location = client.location_path(project_id, compute_region)

my_model = {
    "display_name": model_name,
    "dataset_id": dataset_id,
    "image_classification_model_metadata": {"train_budget": train_budget}
}

response = client.create_model(project_location, my_model)
```

Create model

```
from google.cloud import automl_v1beta1 as automl
client = automl.AutoMLClient()

project_location = client.location_path(project_id, compute_region)

my_model = {
    "display_name": model_name,
    "dataset_id": dataset_id,
    "image_classification_model_metadata": {"train_budget": train_budget}
}

response = client.create_model(project_location, my_model)
```

Create model

```
from google.cloud import automl_v1beta1 as automl
client = automl.AutoMLClient()

project_location = client.location_path(project_id, compute_region)

my_model = {
    "display_name": model_name,
    "dataset_id": dataset_id,
    "image_classification_model_metadata": {"train_budget": train_budget}
}

response = client.create_model(project_location, my_model)
```

Query model

```
from google.cloud import automl_v1beta1 as automl
automl_client = automl.AutoMLClient()
prediction_client = automl.PredictionServiceClient()
```

```
model_full_id = automl_client.model_path(
    project_id, compute_region, model_id
)
```

```
with open(file_path, "rb") as image_file:
    content = image_file.read()
payload = {"image": {"image_bytes": content}}
params = {}
```

```
response = prediction_client.predict(model_full_id, payload, params)
```

Query model

```
from google.cloud import automl_v1beta1 as automl
automl_client = automl.AutoMLClient()
prediction_client = automl.PredictionServiceClient()
```

```
model_full_id = automl_client.model_path(
    project_id, compute_region, model_id
)
```

```
with open(file_path, "rb") as image_file:
    content = image_file.read()
payload = {"image": {"image_bytes": content}}
params = {}
```

```
response = prediction_client.predict(model_full_id, payload, params)
```

Query model

```
from google.cloud import automl_v1beta1 as automl
automl_client = automl.AutoMLClient()
prediction_client = automl.PredictionServiceClient()
```

```
model_full_id = automl_client.model_path(
    project_id, compute_region, model_id
)
```

```
with open(file_path, "rb") as image_file:
    content = image_file.read()
payload = {"image": {"image_bytes": content}}
params = {}
```

```
response = prediction_client.predict(model_full_id, payload, params)
```

Query model

```
from google.cloud import automl_v1beta1 as automl
automl_client = automl.AutoMLClient()
prediction_client = automl.PredictionServiceClient()
```

```
model_full_id = automl_client.model_path(
    project_id, compute_region, model_id
)
```

```
with open(file_path, "rb") as image_file:
    content = image_file.read()
payload = {"image": {"image_bytes": content}}
params = {}
```

```
response = prediction_client.predict(model_full_id, payload, params)
```

cloud.google.com/automl/docs



Playground Prediction Competition

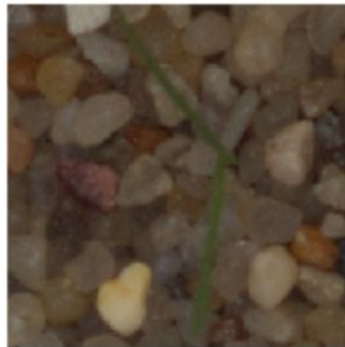
Plant Seedlings Classification

Determine the species of a seedling from an image



Kaggle · 836 teams · 6 months ago

Seedlings



Seedlings

Model	AUC	Mean F-score	Rank
Kaggle Median	N/A	0.9559	418/837
AutoML	0.9895	0.9698	332/837

The Nature
Conservancy



The Nature Conservancy Fisheries Monitoring

Can you detect and classify species of fish?

\$150,000 · 389 teams · a year ago



Google Cloud

Fish



ALB: Albacore tuna (*Thunnus alalunga*)



BET: Bigeye tuna (*Thunnus obesus*)



DOL: Dolphinfish, Mahi Mahi (*Coryphaena hippurus*)



LAG: Opah, Moonfish (*Lampris guttatus*)



SHARK: Various: Silky, Shortfin Mako



YFT: Yellowfin tuna (*Thunnus albacares*)

Fish images are not to scale with one another

Fish



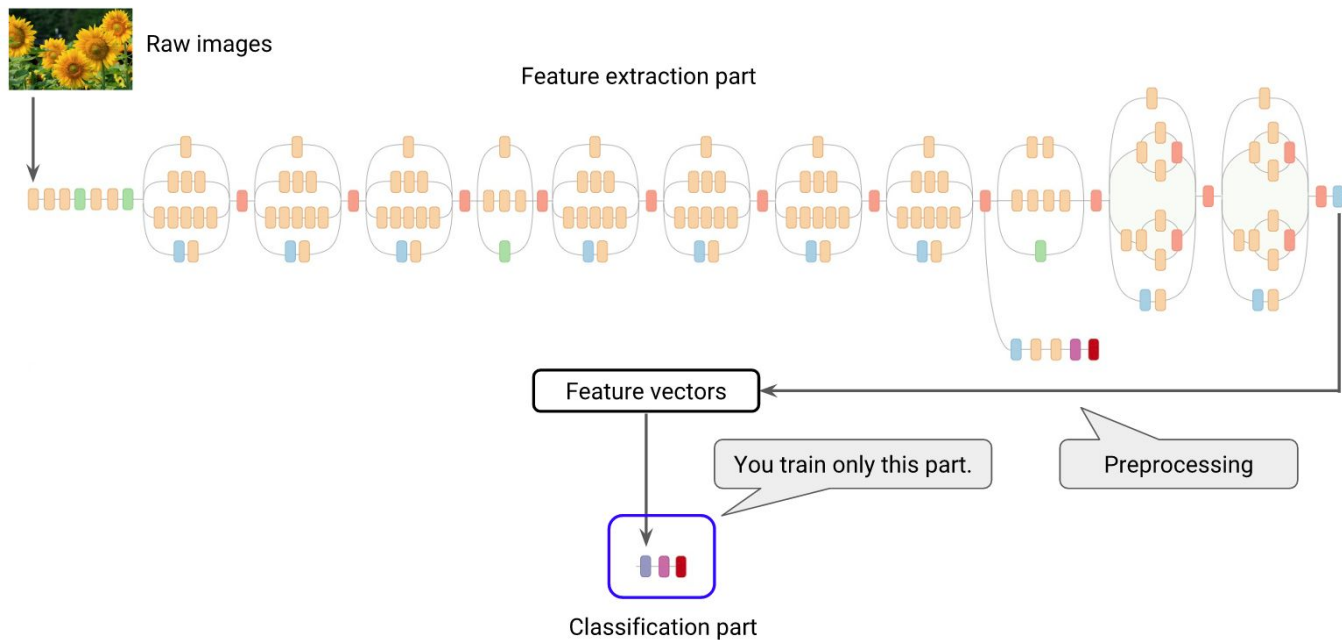
Fish

Model	AUC	Score	Rank
Kaggle Median	N/A	0.9570	144/390
AutoML	0.9849	0.9047	118/390

Behind the Scenes



Transfer Learning



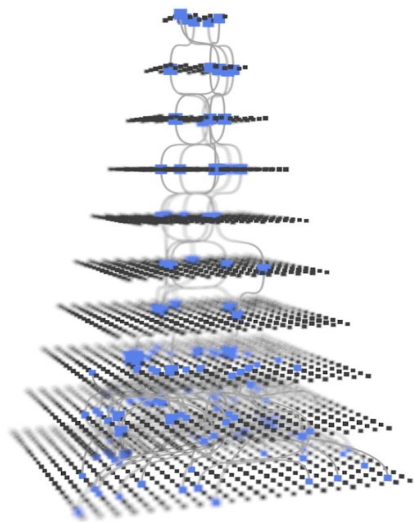
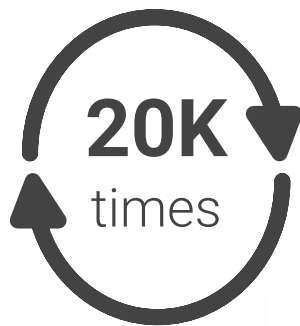
Neural Architecture Search

Neural networks designed by neural networks





Controller: proposes ML models



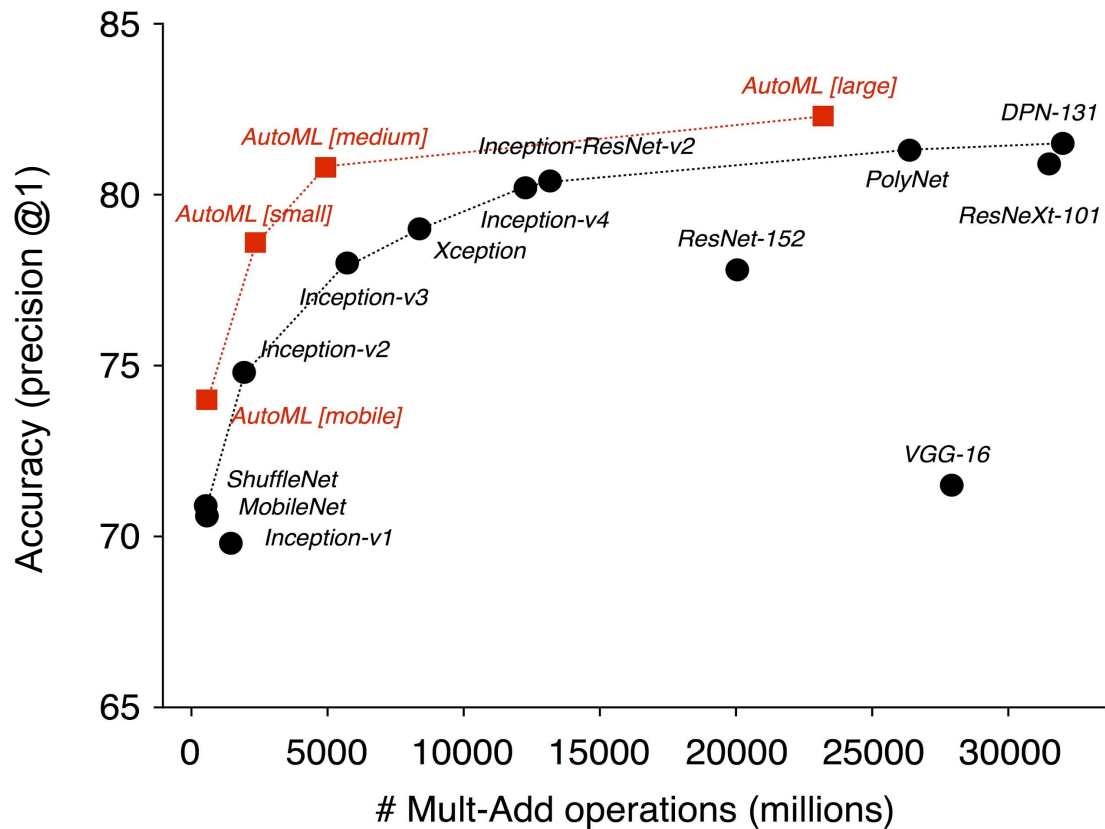
Iterate to
find the
most
accurate
model

Learning Rate →

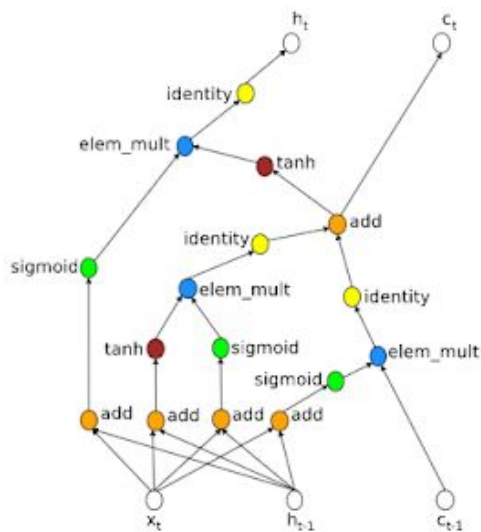
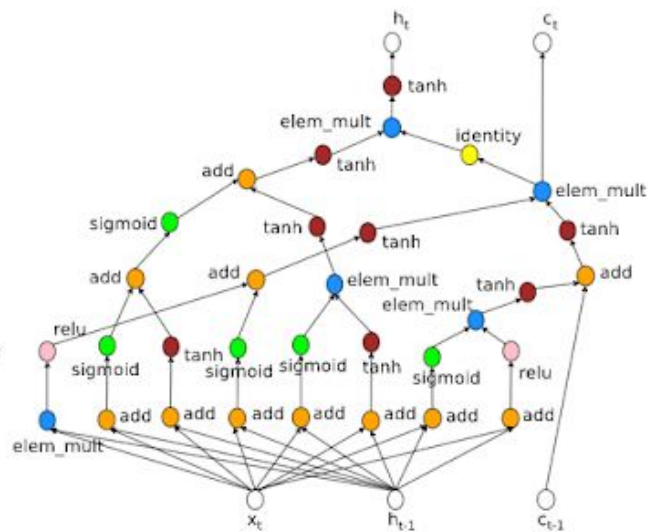
Train & evaluate models



Layers ↓



Neural Architecture Search



Natural Language



How could you use AutoML NL?

- Predict the tag(s) of a Stack Overflow question
- Predict the source of a news headline
- Predict the condition associated with a patient's symptoms

How could you use AutoML NL?

```
var functionName = function() {} vs function functionName() {}
```



I've recently started maintaining someone else's JavaScript code. I'm fixing bugs, adding features and also trying to tidy up the code and make it more consistent.

5562



javascript

- Predict the source of a news headline
- Predict the condition associated with a patient's symptoms

How could you use AutoML NL?

var functionName = function() {} vs function functionName() {}



I've recently started maintaining someone else's JavaScript code. I'm fixing bugs, adding features and also trying to tidy up the code and make it more consistent.

5562



javascript

TECH WE'RE USING

How Technology Is Changing Visual Art



nytimes

- Predict the condition associated with a patient's symptoms

How could you use AutoML NL?

var functionName = function() {} vs function functionName() {}



I've recently started maintaining someone else's JavaScript code. I'm fixing bugs, adding features and also trying to tidy up the code and make it more consistent.

5562



javascript

TECH WE'RE USING

How Technology Is Changing Visual Art



nytimes

Symptoms include fever, chills, muscle aches, cough, congestion, runny nose, headaches, and fatigue.



influenza

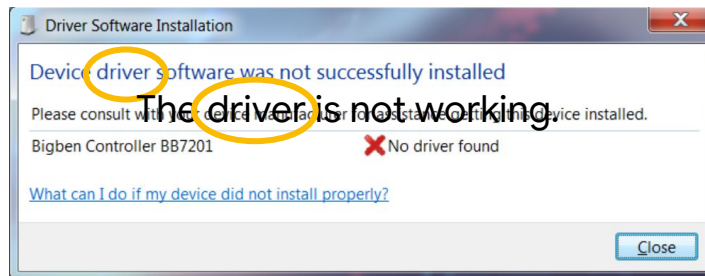
Live demo: AutoML Natural Language

Translation



Translation

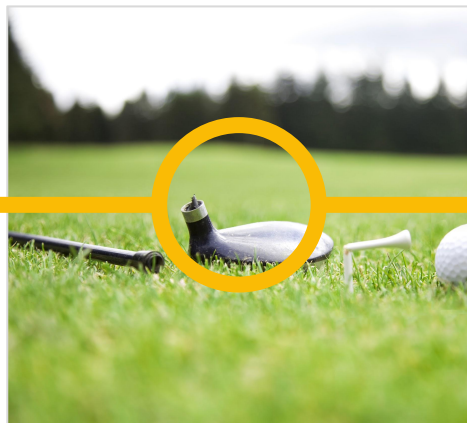
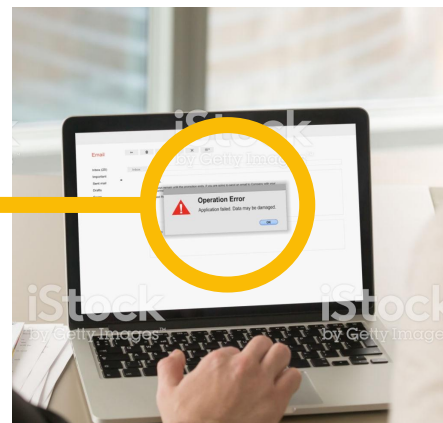
Why is domain-specific translation important?



Translation

Why is domain-specific translation important?

The **driver**
is not working



Live demo: AutoML Translation

European Medicines Agency Docs

Source Language	Target Language	AutoML BLEU	NMT BLEU	Improvement
ES	EN	59.88	51.28	+16.8%
NL	EN	58.82	53.35	+10.3%
FR	EN	54.20	48.26	+12.3%
PT	EN	59.92	52.81	+13.5%
IT	EN	59.70	52.63	+13.4%

Thank You!



+Torry Yang
@torryyang



+Sara Robinson
@SRobTweets