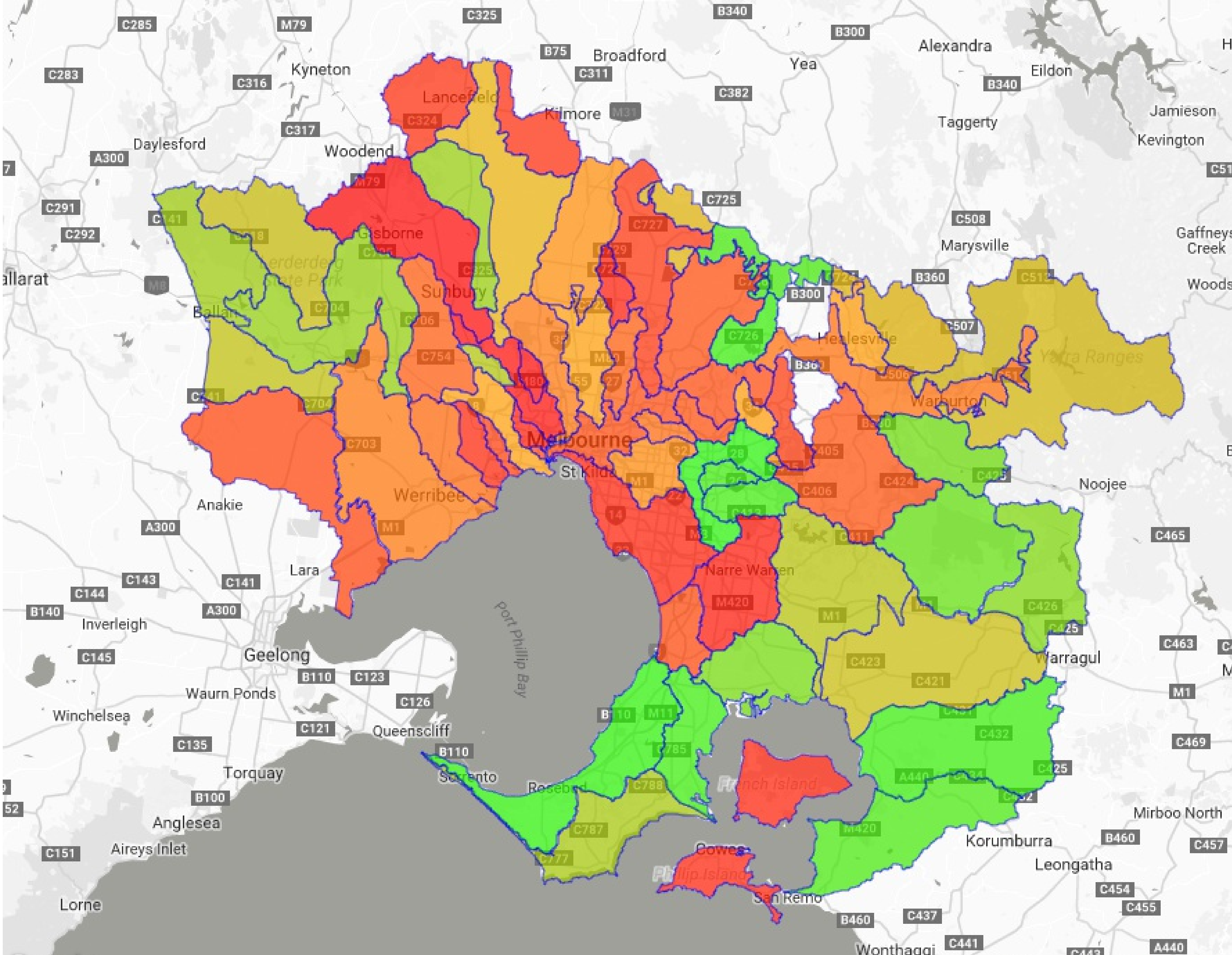
Something from nothing

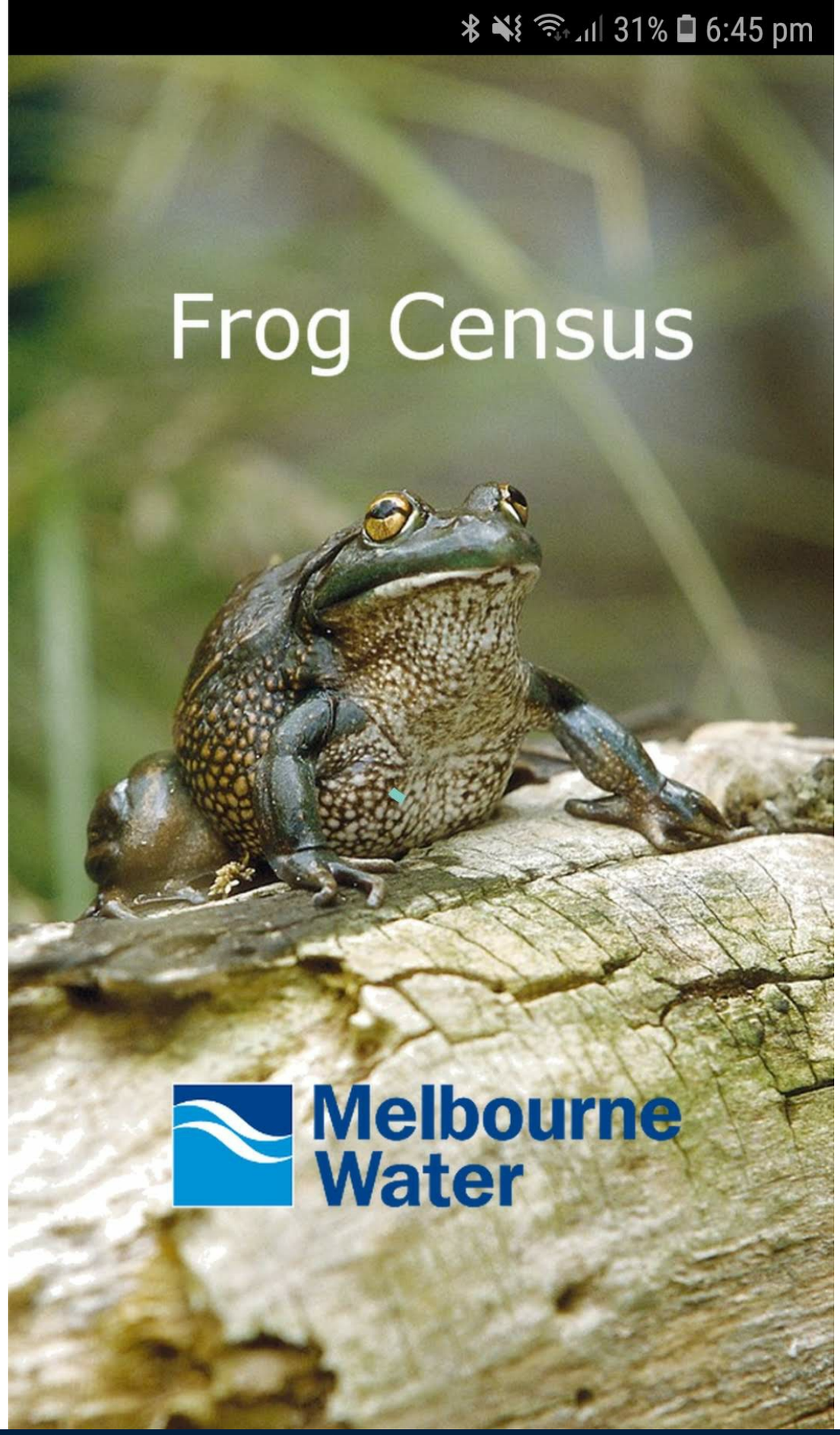
TensorFlow models from scratch

Elizabeth Stark

www.symbolix.com.au

It started with a frog ...


Kyneton
Broadford
Yea
Alexandra
Eildon
Kilmore
Lancefield
Taggerty
Jamieson
Kevington
Daylesford
Woodend
Gisborne
Marysville
Gaffneys Creek
Sunbury
Healesville
Yarra Ranges
Warburton
Melbourne
St Kilda
Noojee
Anakie
Werribee
Lara
Narre Warren
Inverleigh
Geelong
Port Phillip Bay
Warragul
Waurn Ponds
Winchelsea
Queenscliff
Torquay
Sorrento
Rosebud
French Island
Mirboo North
Anglesea
Korumburra
Aireys Inlet
Cowes
Phillip Island
Leongatha
San Remo
Lorne
Wonthaggi

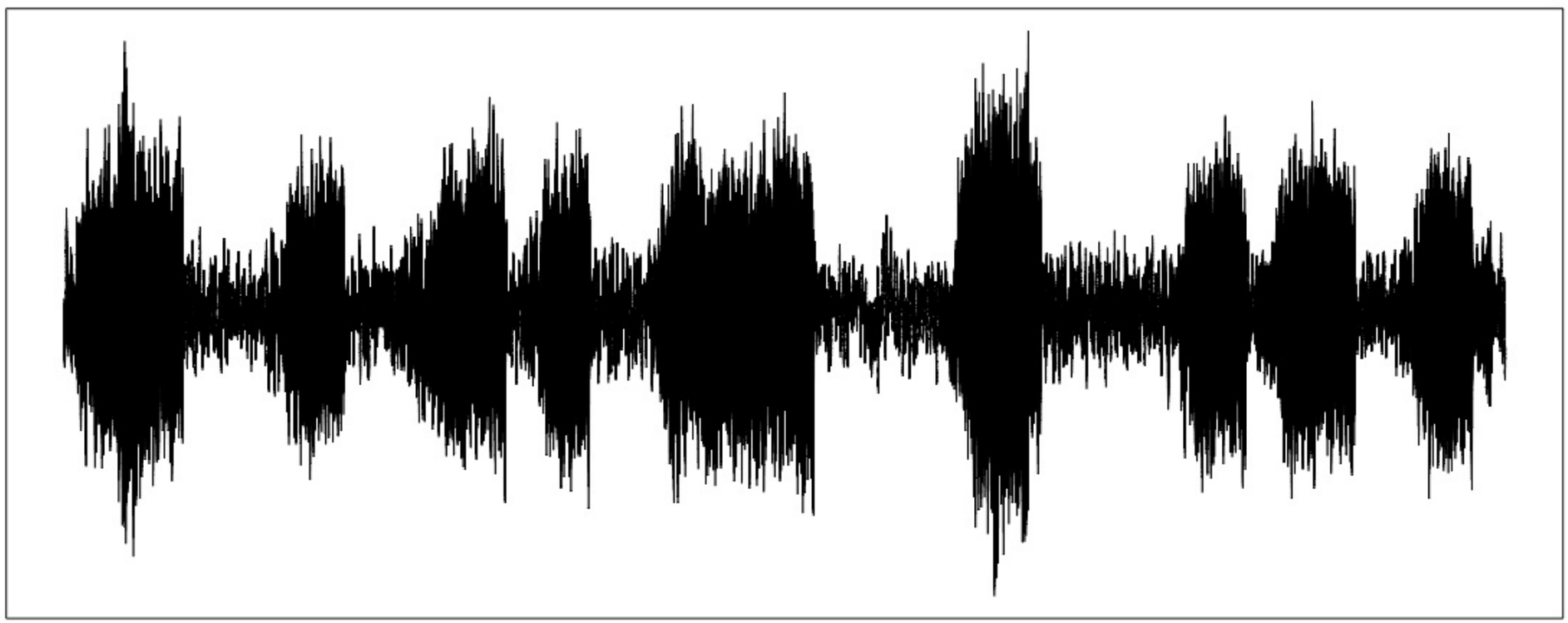

31% 6:45 pm
Frog Census
Melbourne
Water

The first challenge

Solution:

The second challenge

Southern Toadlet Peron's Tree Frog GGF



Computer listening vs computer vision

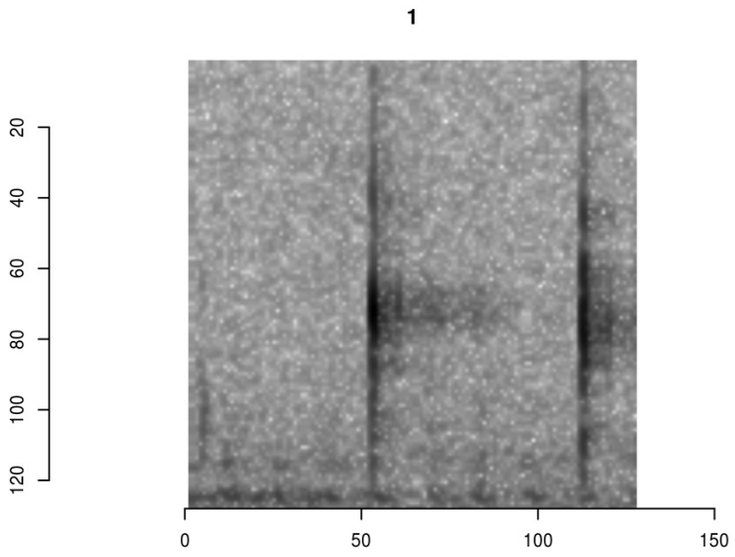
arXiv.org

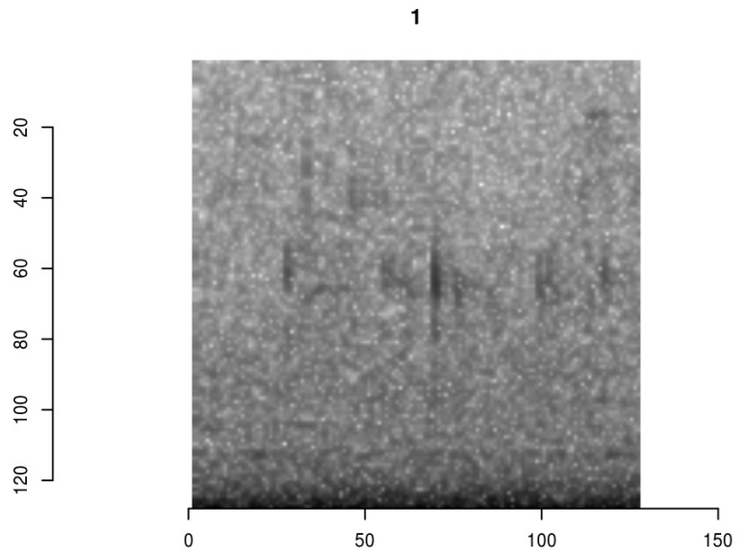
Keyword="sound" : ~100K papers

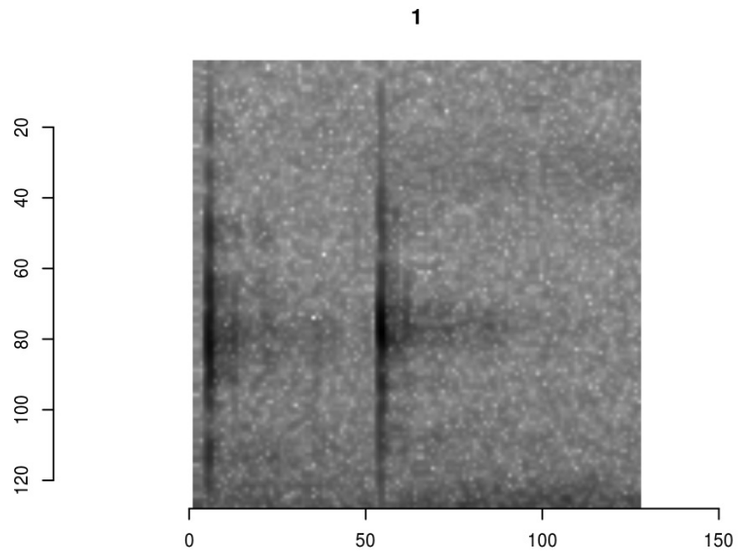
Keyword="image" : ~3.5Mil papers

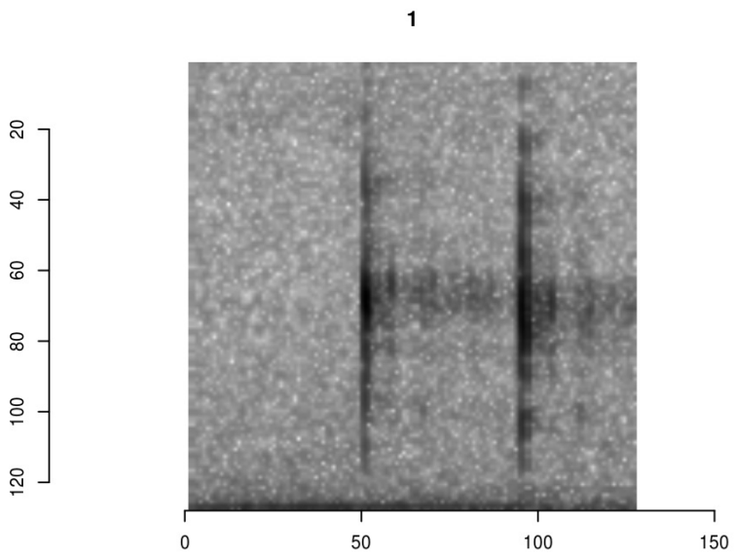
Solution:

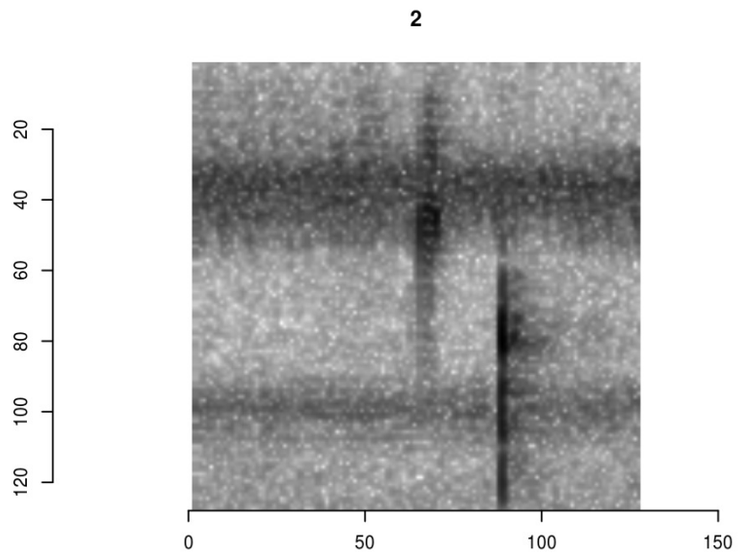
Play with the ~~cool kids toys~~ computer vision tools

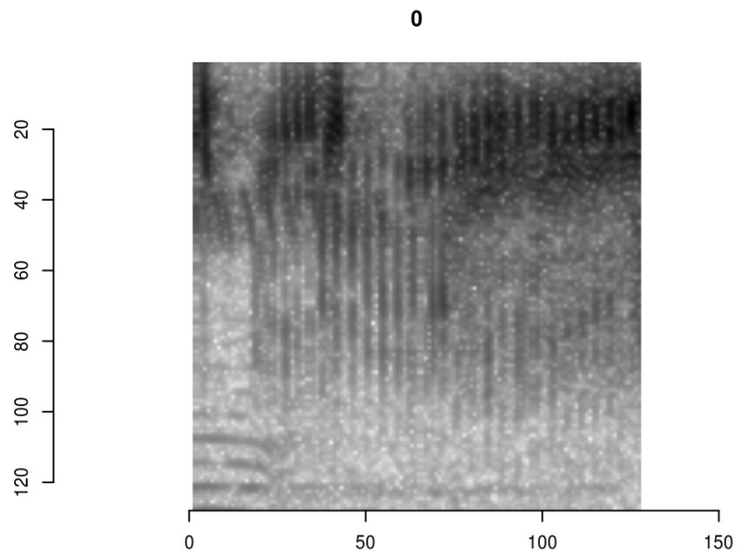

1
20
40
60
80
100
120
0
50
100
150

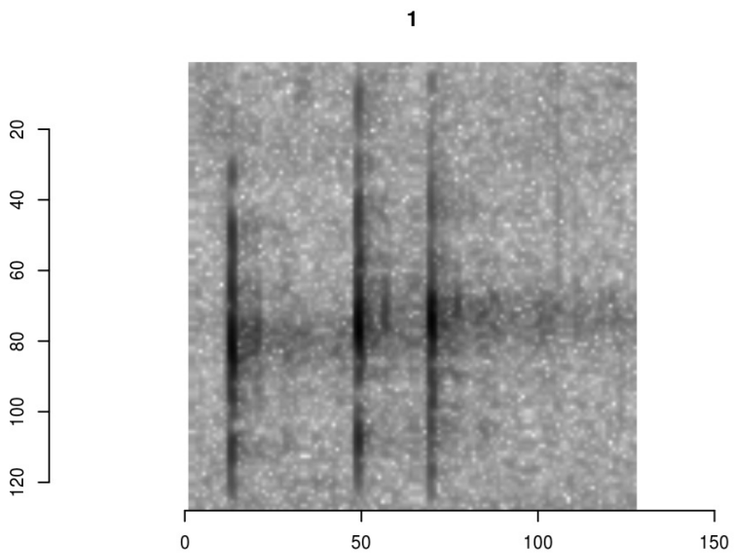

1
20
40
60
80
100
120
0
50
100
150

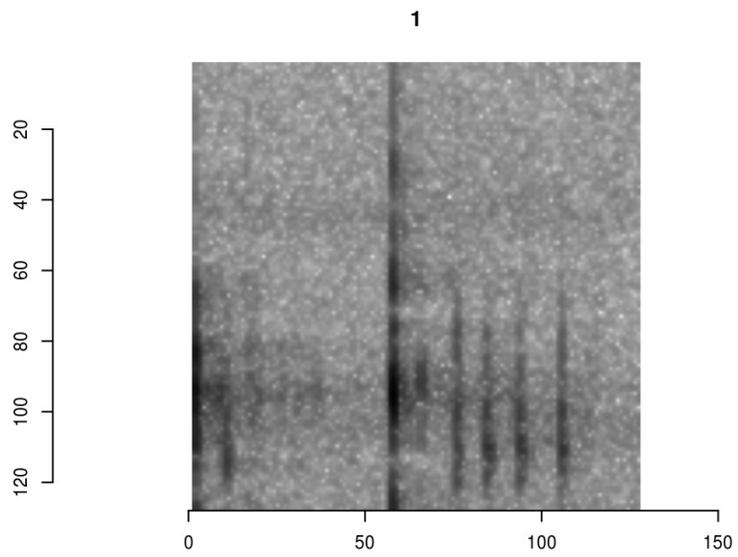

1
20
40
60
80
100
120
0
50
100
150

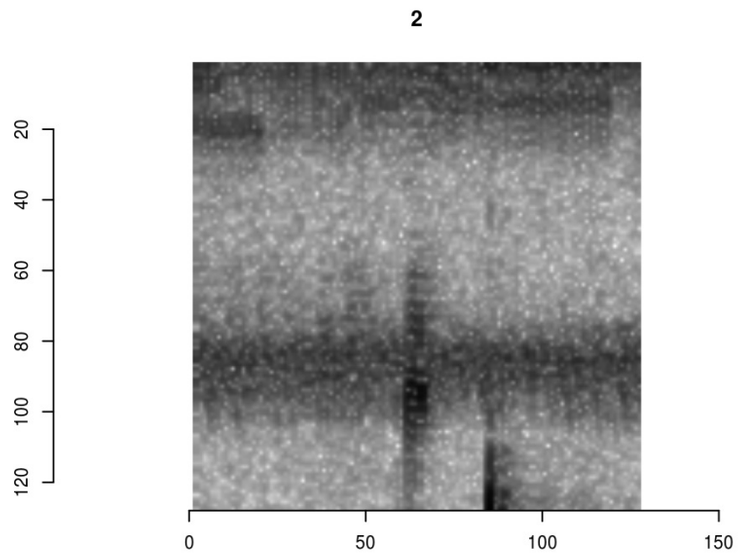

1
20
40
60
80
100
120
0
50
100
150


2
20
40
60
80
100
120
0
50
100
150


0
20
40
60
80
100
120
0
50
100
150


1
20
40
60
80
100
120
0
50
100
150


1
20
40
60
80
100
120
0
50
100
150


2
20
40
60
80
100
120
0
50
100
150

The third challenge

How few data?

- ~1,400 individual tagged calls
- 5 classes
    - Litoria fallax (Eastern Dwarf Tree frog)
    - Limnodynastes tasmanienses (Spotted grass frog)
    - Crinia signifera (Common Eastern froglet)
    - Limnodynastes peronii (Striped Marsh frog)
    - Other frog

..... And about 100 million billion crickets calling over the top

Solution

Our toolbox

~/packages/keras - master - RStudio

Go to file/function | Addins | keras

rs-pooling.R | layers-normalization.R | models.py | mnist_antirectifier.R | mnist_mlp.R | test-cus

Source on Save | Run | Source

```
24 x_train <- x_train / 255
25 x_test <- x_test / 255
26
27 cat(nrow(x_train), 'train samples\n')
28 cat(nrow(x_test), 'test samples\n')
29
30 # convert class vectors to binary class matrices
31 y_train <- to_categorical(y_train, num_classes)
32 y_test <- to_categorical(y_test, num_classes)
33
34 model <- keras_model_sequential()
35 model %>%
36   layer_dense(units = 256, activation = 'relu', input_shape = c(784)) %>%
37   layer_dropout(rate = 0.4) %>%
38   layer_dense(units = 128, activation = 'relu') %>%
39   layer_dropout(rate = 0.3) %>%
40   layer_dense(units = 10, activation = 'softmax')
41
42 summary(model)
43
44 model %>% compile(
45   loss = 'categorical_crossentropy',
46   optimizer = optimizer_rmsprop(),
47   metrics = c('accuracy')
48 )
49
50 history <- model %>% fit(
51   x_train, y_train,
52   batch_size = batch_size,
53   epochs = epochs,
54   verbose = 1,
55   validation_split = 0.2
56 )
```

45:37 (Top Level) R Script

Console | Terminal | Find in Files | R Markdown

~/packages/keras/

```
+ )

> history <- model %>% fit(
+   x_train, y_train,
+   batch_size = batch_size,
+   epochs = epochs,
+   verbose = 1,
+   validation_split = 0.2
+ )
Train on 48000 samples, validate on 12000 samples
Epoch 1/30
48000/48000 [==============================] - 2s - loss: 0.4333 - acc: 0.8675 - val_loss: 0.1725 - val_acc:
0.9495
Epoch 2/30
48000/48000 [==============================] - 2s - loss: 0.2037 - acc: 0.9401 - val_loss: 0.1240 - val_acc:
0.9622
Epoch 3/30
48000/48000 [==============================] - 2s - loss: 0.1602 - acc: 0.9531 - val_loss: 0.1065 - val_acc:
0.9690
Epoch 4/30
19072/48000 [==========>...................] - ETA: 1s - loss: 0.1324 - acc: 0.9610
```

Environment | History | Connections | Build | Git

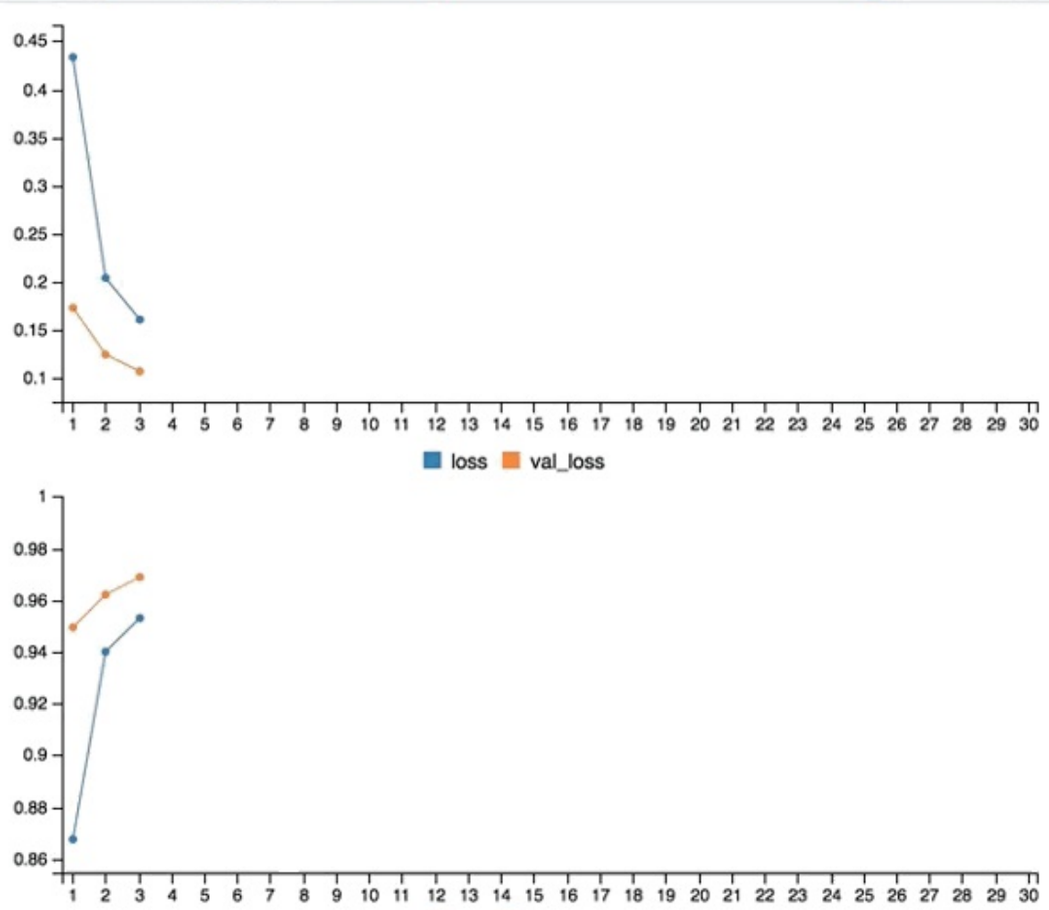
Import Dataset | List

Global Environment

Data	
history	List of 2
mnist	Large list (2 elements, 209.6 Mb)
score	List of 2
x_test	Large matrix (7840000 elements, 59.8 Mb)
x_train	Large matrix (47040000 elements, 358.9 Mb)
y_test	Large matrix (100000 elements, 781.4 Kb)
y_train	Large matrix (600000 elements, 4.6 Mb)
Values	
batch_size	128
epochs	30
model	Model
num_classes	10

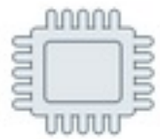
Files | Plots | Packages | Help | Viewer

Zoom | Export | Publish



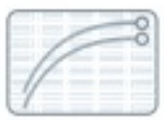
With all the usual tools

GPUs



CloudML

TensorBoard



Training Flags

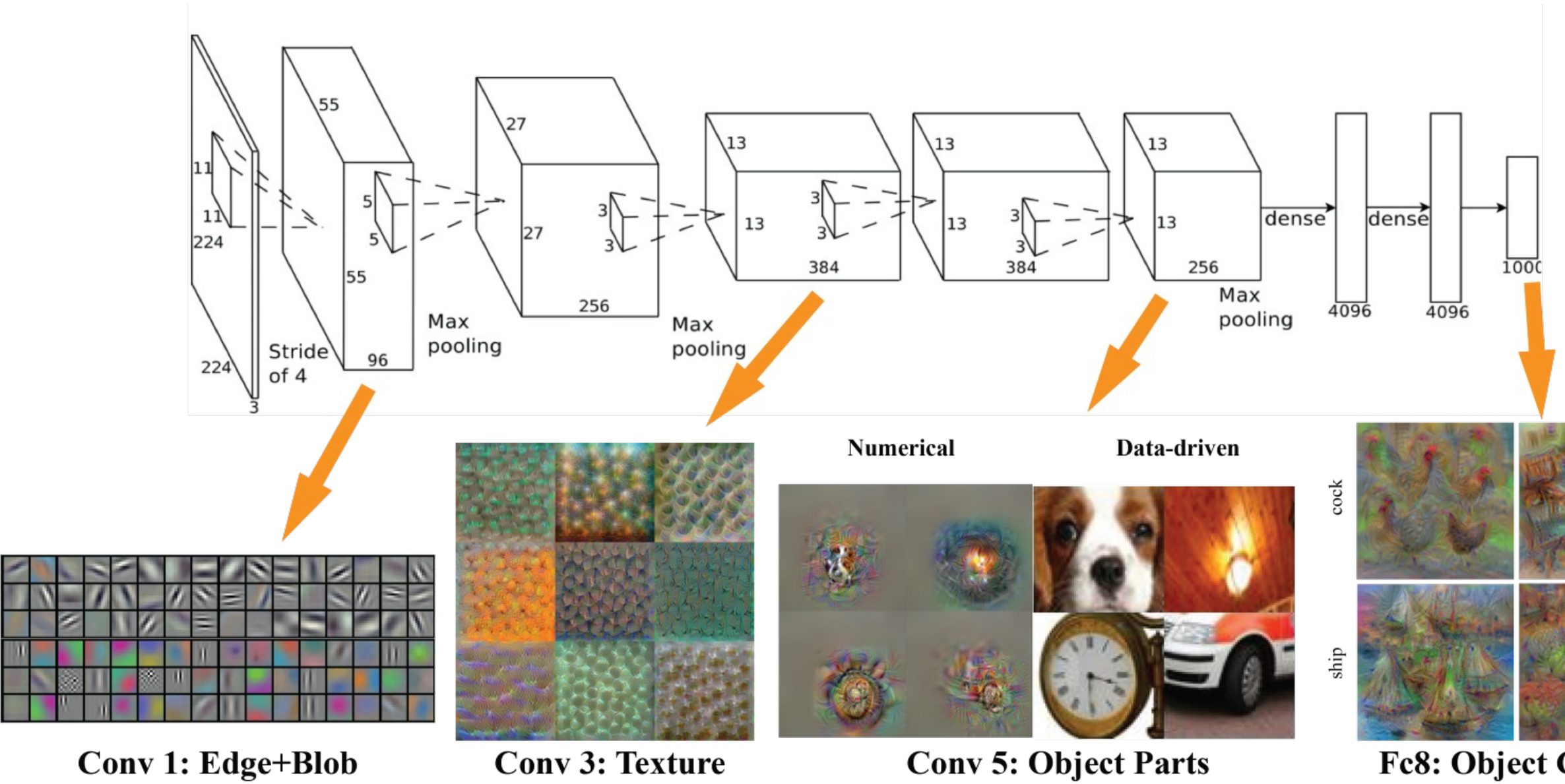
Datasets API

Training Runs

Deployment

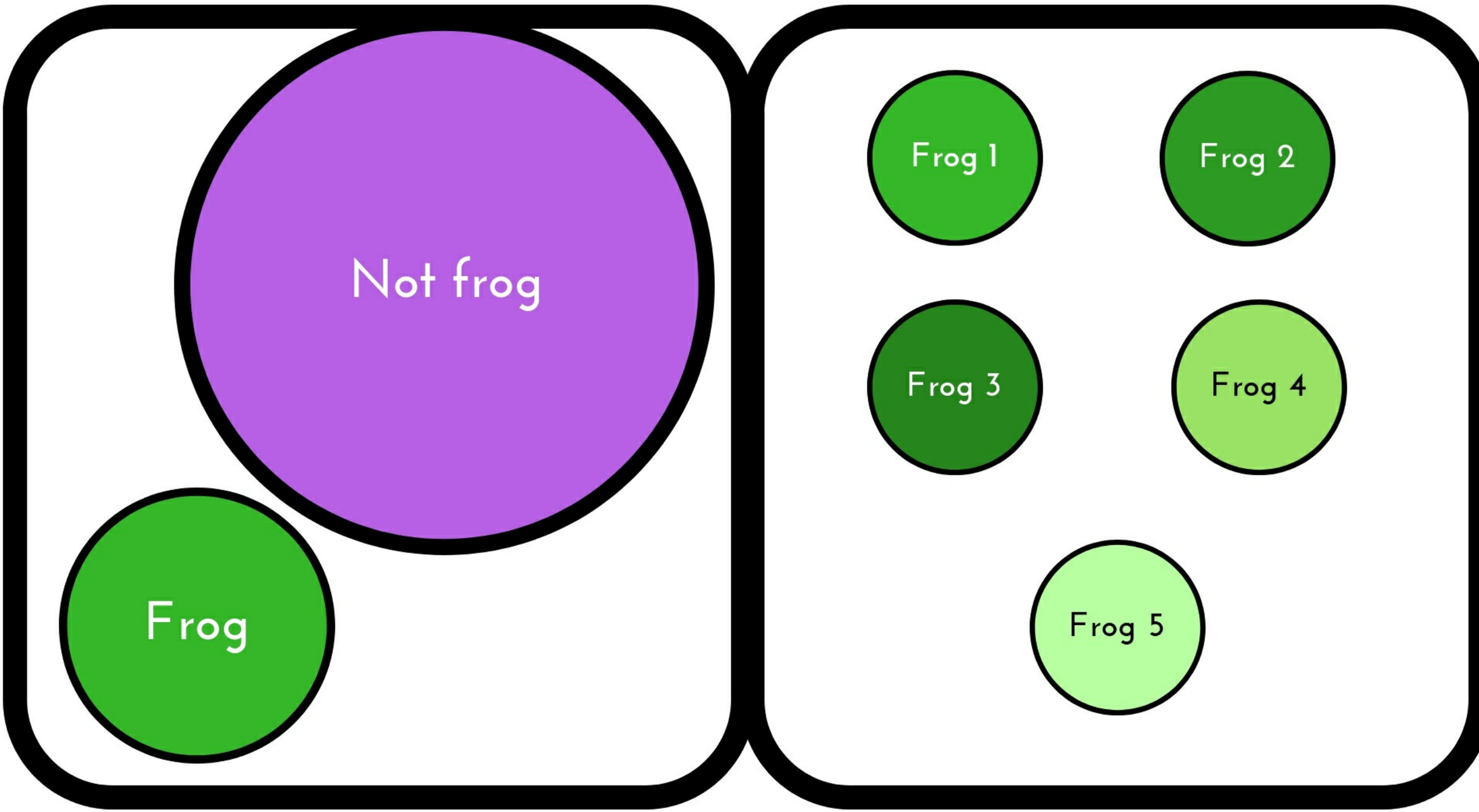
Tricks and tips

Leave the deepest networks for the big datasets



Conv 1: Edge+Blob **Conv 3: Texture** **Conv 5: Object Parts** **Fc8: Object Classes**

Even it out


Not frog
Frog
Frog 1
Frog 2
Frog 3
Frog 4
Frog 5

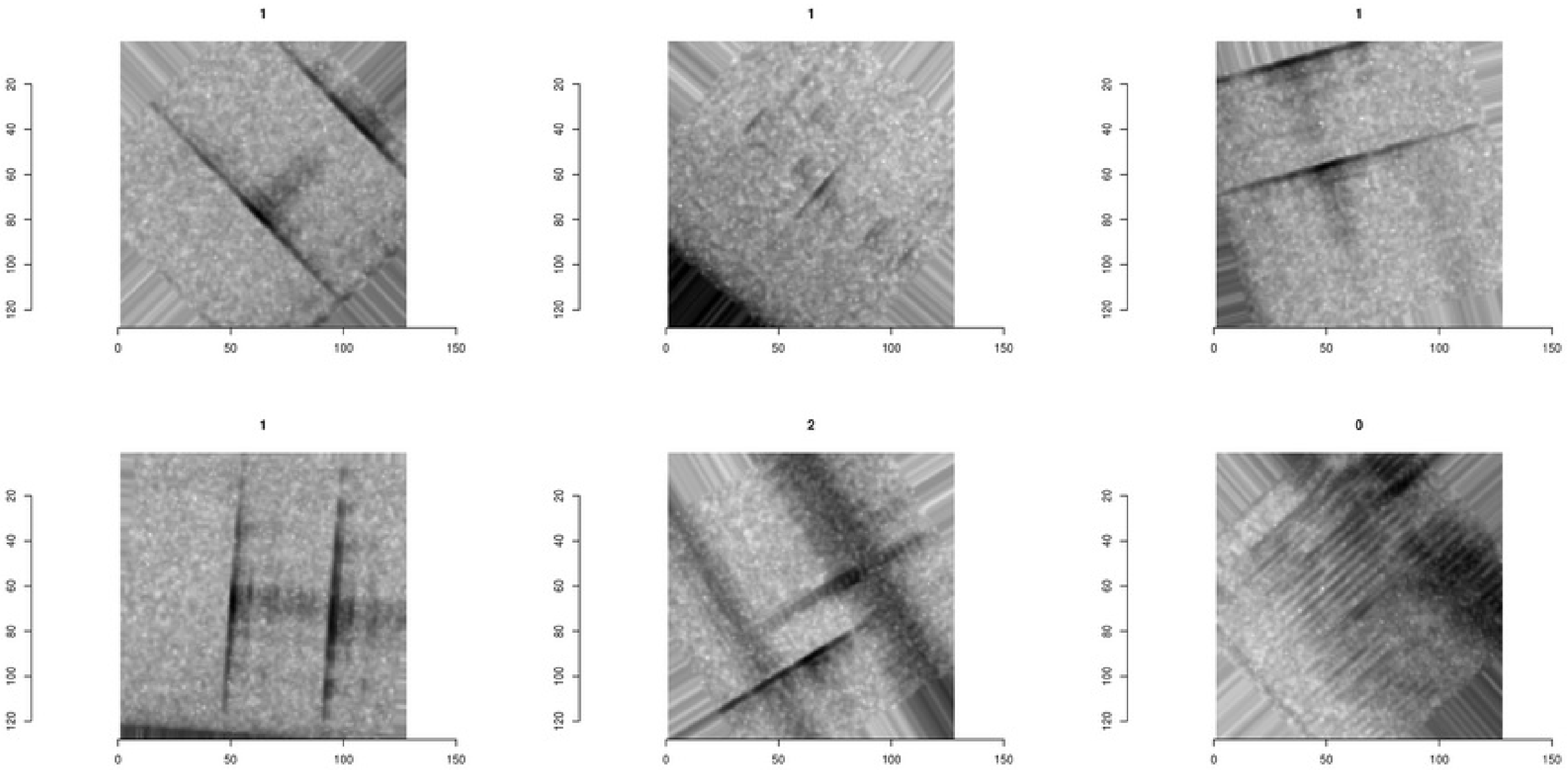
Data augmentation

- Normalise/standardise
- Rotate
- Shift/Jitter
- Shear
- Zoom
- Flip

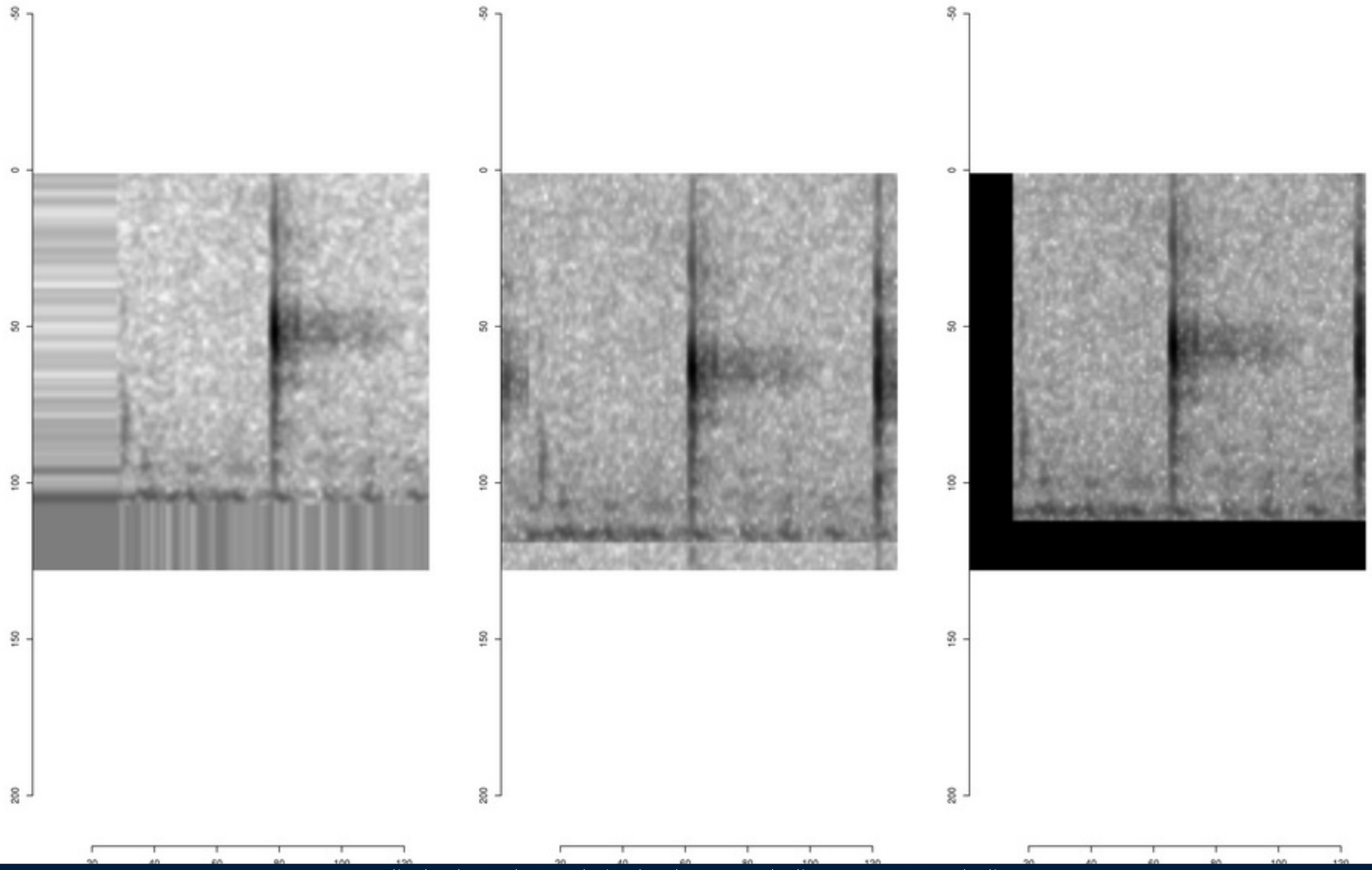
Rotation

```
data_gen_aug <- image_data_generator(
  rescale = 1,
  featurewise_center = FALSE,
  featurewise_std_normalization = FALSE,
  samplewise_center = TRUE,
  samplewise_std_normalization = TRUE,
  rotation_range = 90    # in degrees
  )

batch_aug_rotate <- augBatch(object = data_gen_aug, img = img, lbl = lbl)
```

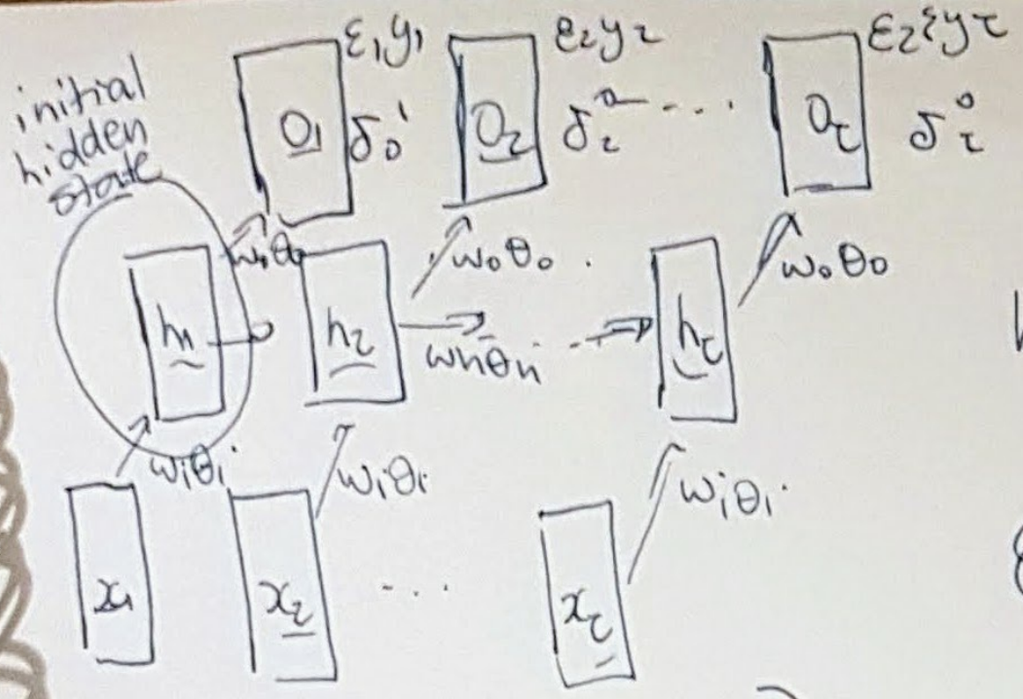


Shift



Sometimes you need the big guns

BACKPROP : UNROLL THE NETWORK FOR τ LENGTH

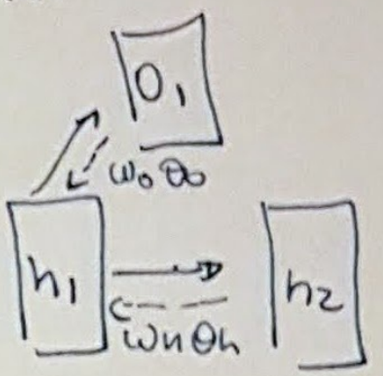


$$x = \langle x_1\ x_2 \cdots x_\tau \rangle$$

$$h_0 = (0, 0, 0 \cdots) \quad (m)$$

$$\mathcal{E} = \sum_{i=1}^{\tau} e_i \rightarrow \text{just add errors.}$$

$$\text{eg } e_i = \Sigma \|y_i - O_i\|^2$$



Compute the δ's

$$\delta_t^o = \frac{\partial f^o}{\partial \varepsilon_t} \odot \frac{\partial \varepsilon_t}{\partial e} \quad \forall t \in [1 \cdots \tau]$$

$$\delta_1^h = \frac{\partial f^h}{\partial e} \odot \left(w_o \delta_1^o + w_h \delta_2^h \right)$$

$$\delta_t^h = \frac{\partial f^h}{\partial e} \odot \left(\underbrace{w_o \delta_t^o}_{\text{backprop from output}} + \underbrace{w_h \delta_{t+1}^h}_{\text{from next hidden state } t+1} \right)$$

Then

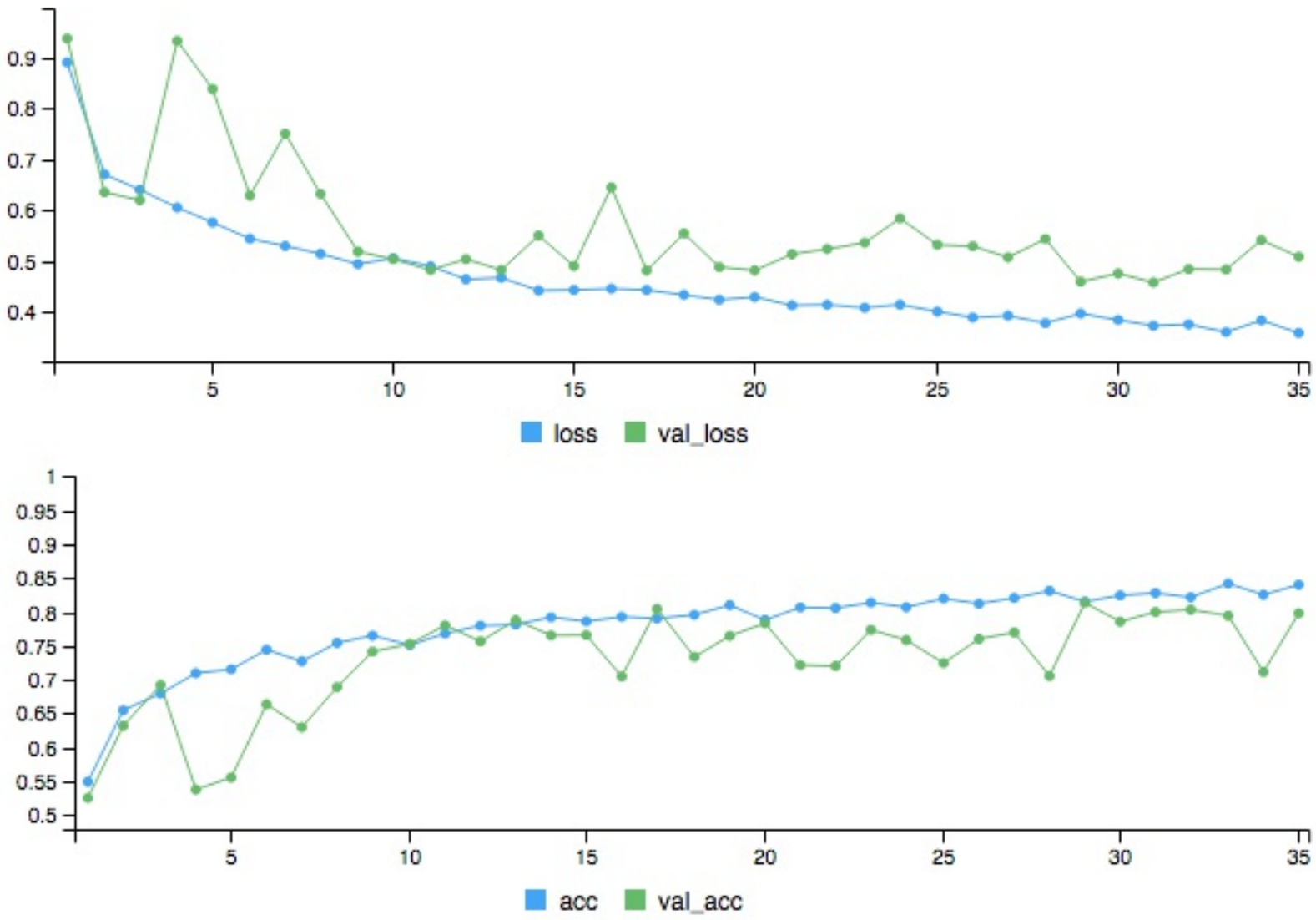
$$\nabla w_o = \sum_{t=1}^{\tau} h_t \cdot (\delta_t^o)^T$$

Bidirectional ~~LSTMS~~. RNNS

The result?

Training Run SUMMARY OUTPUT CODE 2018-05-08T02-57-56Z

History



Run

context	local
script	Conv4Dense2_asymconv.R
started	2018-05-08 02:57:56 GMT
time	00:11:52

Metrics

loss	0.3590
acc	0.8403
val_loss	0.5083
val_acc	0.7981

Flags

model_clayers	3
conv_activation	relu
dense_activation	relu
ksz1	5
ksz2	3
mp1	2

Next steps

symbolix