SudokuSolver_P2

March 27, 2022

0.1 # Building a Sudoku Solver Part 2: Deep Neural Networks

A quick recap of the steps needed to do solve sudoku from images:

1. Finding the largest square in the image
2. Unskewing the image
3. Extracting the cells
4. Deciding if each cell contains a digit or not
5. Applying an ML model like CNN or a simple neural network to find the digits
6. Solving the sudoku

Steps 1-3 use out-of-the-box well-defined computer vision methods. We used logistic regression for step 4 here and it worked great! But for step 5, it only produced a 57% accuracy, meaning around half the time the digits were misclassified.

How do we make a model that is more accurate in classifying the digits present in the sudoku cells that we extract?

> We can try using neural networks, a class of models so powerful that they can allow us to skip step 4, directly classyfing the digit present (1-9) and if there is no digit at all.

1 Setup code:

The code below basically takes the sudoku images, applies step 1-3 and gives us a dataset consisting of 32x32 pixel images of the sudoku cells, having either the digit 1-9 or no digit at all (let's consider this label as 0)

1.1 Installing required packages

[1]:
```
!pip install wget
```

```
Collecting wget
  Downloading wget-3.2.zip (10 kB)
Building wheels for collected packages: wget
  Building wheel for wget (setup.py) … done
  Created wheel for wget: filename=wget-3.2-py3-none-any.whl size=9675
sha256=9ee6cef0e87bcf9efb2939a473a28b9b241d79304ca29cdf3643970c44885a1b
  Stored in directory: /root/.cache/pip/wheels/a1/b6/7c/0e63e34eb06634181c63adac
```

```
ca38b79ff8f35c37e3c13e3c02
Successfully built wget
Installing collected packages: wget
Successfully installed wget-3.2
```

1.2 Importing required packages

[2]:
```
import cv2
import numpy as np
import wget
import bz2
import random

from glob import glob
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.linear_model import LogisticRegression
from tqdm.auto import tqdm
import keras
import tensorflow
print(keras.__version__)
print(tensorflow.__version__)
```

```
2.8.0
2.8.0
```

1.3 Download the zip files

[3]:
```
train_zip = wget.download('https://github.com/jeffreywolberg/sudoku_dataset/
 ↪blob/master/datasets/v2_train.tar.bz2?raw=true')
test_zip = wget.download('https://github.com/jeffreywolberg/sudoku_dataset/blob/
 ↪master/datasets/v2_test.tar.bz2?raw=true')
csv_data = wget.download('https://raw.githubusercontent.com/jeffreywolberg/
 ↪sudoku_dataset/master/outlines_sorted.csv')
```

1.4 Extract and move the zip files to Data folder

[4]:
```
!if [ -d '/content/Data/' ]; then rm -rf /content/Data else ; fi
!mkdir Data
!tar -xf /content/v2_train.tar.bz2 -C /content/Data/ && rm -rf /content/
 ↪v2_train.tar.bz2
!tar -xf /content/v2_test.tar.bz2 -C /content/Data/  && rm -rf /content/v2_test.
 ↪tar.bz2
```

```
!mv outlines_sorted.csv /content/Data
```

1.5 Helper functions

[5]:
```
def largest4SideContour(image):
    contours, h = cv2.findContours(
        image, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

    # Sorting contours in descending order based on the area
    contours = sorted(contours, key=cv2.contourArea, reverse=True)

    # Looking at top 5 contours and checking if any of them are of size 4
    for cnt in contours[:min(5, len(contours))]:
        if len(approx(cnt)) == 4:
            return cnt
    return None


def approx(cnt):
    try:
        peri = cv2.arcLength(cnt, True)
        app = cv2.approxPolyDP(cnt, 0.01 * peri, True)
        return app
    except:
        return None


def get_rectangle_corners(cnt):
    pts = cnt.reshape(4, 2)
    rect = np.zeros((4, 2), dtype="float32")

    # the top-left point has the smallest sum whereas the
    # bottom-right has the largest sum
    s = pts.sum(axis=1)
    rect[0] = pts[np.argmin(s)]
    rect[2] = pts[np.argmax(s)]

    # compute the difference between the points -- the top-right
    # will have the minumum difference and the bottom-left will
    # have the maximum difference
    diff = np.diff(pts, axis=1)
    rect[1] = pts[np.argmin(diff)]
    rect[3] = pts[np.argmax(diff)]
    return rect
```

```
def warp_perspective(rect, grid):
    (tl, tr, br, bl) = rect
    widthA = np.sqrt(((br[0] - bl[0]) ** 2) + ((br[1] - bl[1]) ** 2))
    widthB = np.sqrt(((tr[0] - tl[0]) ** 2) + ((tr[1] - tl[1]) ** 2))

    # ...and now for the height of our new image
    heightA = np.sqrt(((tr[0] - br[0]) ** 2) + ((tr[1] - br[1]) ** 2))
    heightB = np.sqrt(((tl[0] - bl[0]) ** 2) + ((tl[1] - bl[1]) ** 2))

    # take the maximum of the width and height values to reach
    # our final dimensions
    maxWidth = max(int(widthA), int(widthB))
    maxHeight = max(int(heightA), int(heightB))

    # construct our destination points which will be used to
    # map the screen to a top-down, "birds eye" view
    dst = np.array([
        [0, 0],
        [maxWidth - 1, 0],
        [maxWidth - 1, maxHeight - 1],
        [0, maxHeight - 1]], dtype="float32")

    # calculate the perspective transform matrix and warp
    # the perspective to grab the screen
    M = cv2.getPerspectiveTransform(rect, dst)
    warp = cv2.warpPerspective(grid, M, (maxWidth, maxHeight))
    return cv2.resize(warp, (288, 288))


def getTopLine(image):
    for i, row in enumerate(image):
        if np.any(row):
            return i
    return None


def getBottomLine(image):
    for i in range(image.shape[0] - 1, -1, -1):
        if np.any(image[i]):
            return i
    return None


def getLeftLine(image):
    for i in range(image.shape[1]):
        if np.any(image[:, i]):
            return i
```

```
    return None


def getRightLine(image):
    for i in range(image.shape[1] - 1, -1, -1):
        if np.any(image[:, i]):
            return i
    return None


def rowShift(image, start, end, length):
    shifted = np.zeros(image.shape)
    if start + length < 0:
        length = -start
    elif end + length >= image.shape[0]:
        length = image.shape[0] - 1 - end

    for row in range(start, end + 1):
        shifted[row + length] = image[row]
    return shifted


def colShift(image, start, end, length):
    shifted = np.zeros(image.shape)
    if start + length < 0:
        length = -start
    elif end + length >= image.shape[1]:
        length = image.shape[1] - 1 - end

    for col in range(start, end + 1):
        shifted[:, col + length] = image[:, col]
    return shifted


def centerX(digit):
    topLine = getTopLine(digit)
    bottomLine = getBottomLine(digit)
    if topLine is None or bottomLine is None:
        return digit
    centerLine = (topLine + bottomLine) >> 1
    imageCenter = digit.shape[0] >> 1
    digit = rowShift(
        digit, start=topLine, end=bottomLine, length=imageCenter - centerLine)
    return digit


def centerY(digit):
```

```
    leftLine = getLeftLine(digit)
    rightLine = getRightLine(digit)
    if leftLine is None or rightLine is None:
        return digit
    centerLine = (leftLine + rightLine) >> 1
    imageCenter = digit.shape[1] >> 1
    digit = colShift(
        digit, start=leftLine, end=rightLine, length=imageCenter - centerLine)
    return digit
```

1.6 More helper functions..

[6]:
```
def process_image(path: str):
    # Load image
    try:
        img = cv2.imread(path)

        img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        img = cv2.adaptiveThreshold(img.astype(np.uint8), 255, cv2.
 ↪ADAPTIVE_THRESH_MEAN_C,
                                    cv2.THRESH_BINARY, 11, 3)
        img = 255 - img
        kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (2, 2))
        img = cv2.morphologyEx(img, cv2.MORPH_CLOSE, kernel)

        contours, h = cv2.findContours(img, cv2.RETR_TREE, cv2.
 ↪CHAIN_APPROX_SIMPLE)
        max_contour = max(contours, key=cv2.contourArea)
        x, y, w, h = cv2.boundingRect(max_contour)
        img = img[y:y + h, x:x + w]
        img = cv2.resize(img, (288, 288), interpolation=cv2.INTER_AREA)
        img2 = img.copy()
        largest = largest4SideContour(img2)
        app = approx(largest)
        if app is None:
            return img
        corners = get_rectangle_corners(app)
        img = warp_perspective(corners, img)
        return img
    except Exception as e:
        print(str(e))
        return None

def ground_truth(path: str):
    with(open(path, 'r')) as fin:
```

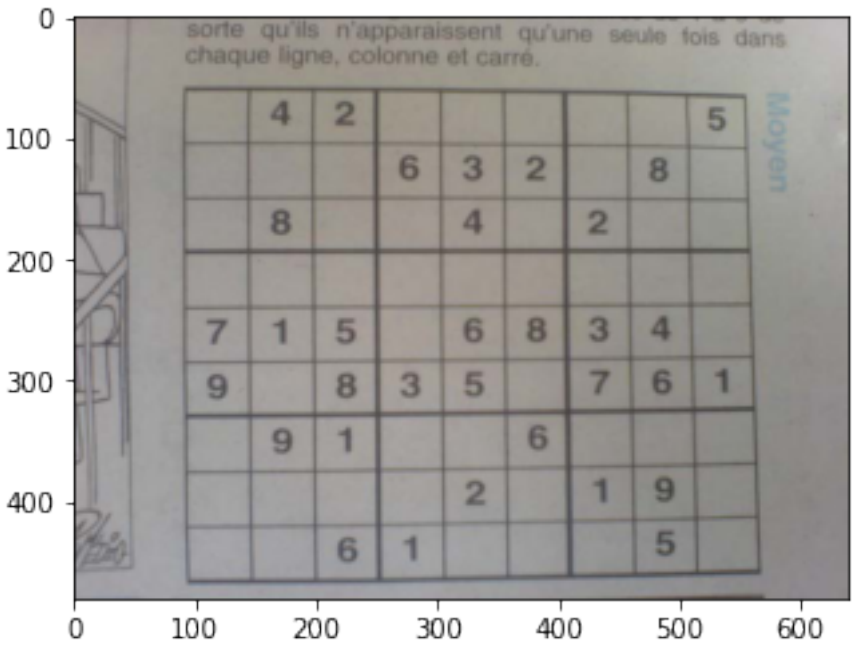
```
        lines = fin.readlines()
        truth = []

        # First line contains the details of phone
        # Second line contains size of source image
        # Skipping both these lines
        for line in lines[2:]:
            for value in line.split():
                truth.append(int(value))

        return np.array(truth) # > 0


def clean(cell):
    kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (2, 2))
    cell = cv2.morphologyEx(cell, cv2.MORPH_CLOSE, kernel)
    return cell


def centerDigit(digit):
    digit = centerX(digit)
    digit = centerY(digit)
    return digit


def get_cells(sudoku, size=288):
    sudoku = sudoku / 255.
    cells = []
    cell_shape = size // 9
    for i in range(9):
        i = cell_shape * i
        for j in range(9):
            j = j * cell_shape
            cell = sudoku[i: i+cell_shape, j: j+cell_shape]
            cell = clean(cell)
            cell = centerDigit(cell)
            cells.append(cell)

    return np.array(cells)


def compare_outputs(pred, truth):
    errors = 0
    for i in range(len(pred)):
        if pred[i] != truth[i]:
            errors += 1
            break
```

```
    if errors:
        return 0
    return 1


def get_sudoku(path: str, model):
    sudoku = process_image(path)

    if sudoku is None:
        raise Exception

    cells = get_cells(sudoku, 288)
    cells = np.reshape(cells, (cells.shape[0], 32, 32, 1))

    # for cell in cells:
    #   cell = cell.reshape((32, 32))
    #   plt.imshow(cell)
    #   plt.show()
    #   v_pred = model.predict(cell.reshape((1, 32, 32, 1)))
    #   print('Predicted: ',np.argmax(v_pred, axis=1))
    #   input("Continue")

    v_pred = model.predict(cells)
    v_pred = np.argmax(v_pred, axis=1)
    sudoku_ext = np.reshape(v_pred, (9, 9))
    return sudoku_ext
```

2 Image before pre-processing

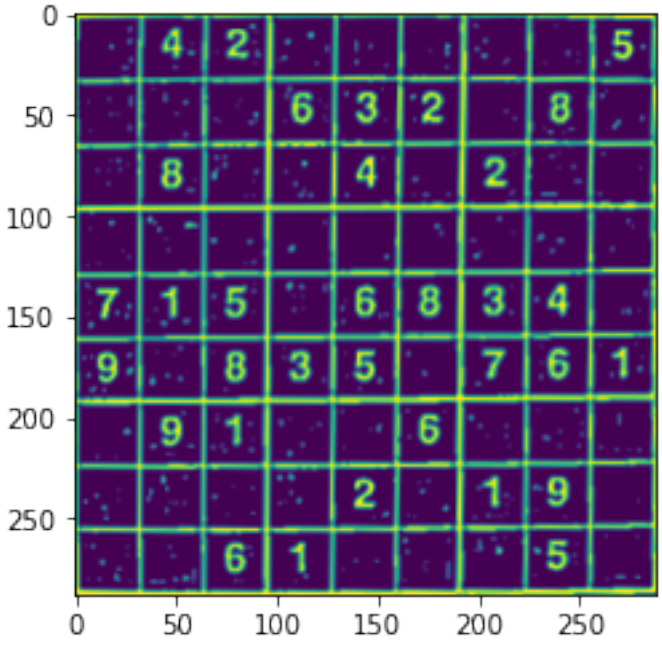
[7]:
```
import matplotlib.pyplot as plt
```

[8]:
```
img = plt.imread('/content/Data/v2_train/image10.jpg')
plt.imshow(img)
plt.show()
```



2.1 Image after pre-processing

[9]:
```
img = process_image('/content/Data/v2_train/image10.jpg')
plt.imshow(img)
plt.show()
```



[10]:
```
img.shape
```

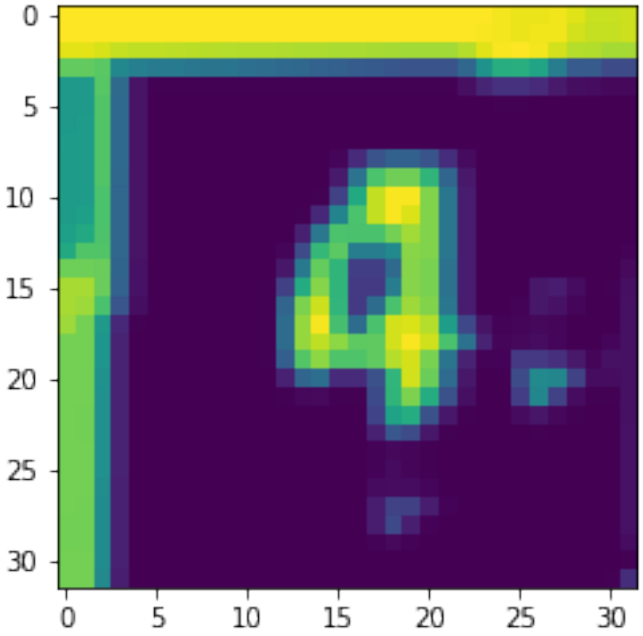
[10]:
```
(288, 288)
```

[11]:
```
cells_img = get_cells(img, size=288)
```

[12]:
```
cells_img.shape
```

[12]:
```
(81, 32, 32)
```

[13]:
```
plt.imshow(cells_img[1])
plt.show()
```



3 Preparing the training dataset

[14]:
```
X_train = []
y_train = []
y_train_all = []

for path in tqdm(glob('/content/Data/v2_train/*.jpg')):
  path = path[:-4]
```

```
  processed_img = process_image(path + '.jpg')
  if processed_img is None:
    continue
  X_train.append(processed_img)
  y_train.append(ground_truth(path + '.dat') > 0)
  y_train_all.append(ground_truth(path + '.dat'))

X_train = np.array(X_train)
y_train = np.array(y_train)
y_train_all = np.array(y_train_all)

X_test = []
y_test = []
y_test_all = []

for path in tqdm(glob('/content/Data/v2_test/*.jpg')):
  path = path[:-4]
  processed_img = process_image(path + '.jpg')
  if processed_img is None:
    continue
  X_test.append(processed_img)
  y_test.append(ground_truth(path + '.dat') > 0)
  y_test_all.append(ground_truth(path + '.dat'))

X_test = np.array(X_test)
y_test = np.array(y_test)
y_test_all = np.array(y_test_all)
```

```
  0%|          | 0/160 [00:00<?, ?it/s]
```

```
  0%|          | 0/40 [00:00<?, ?it/s]
```

- The 288 x 288 image here is the entire 9x9 sudoku grid, giving us 81 different 32x32 cells either containing a digit from 1-9 or no digit at all.
- There are 160 such images for training and 40 for testing. Leading to 12960 (160 * 81) cells to train our model to detect what the cell has and 3240 (40 * 81) cells to validate our model on.

[15]:
```
X_train.shape, X_test.shape
```

[15]:
```
((160, 288, 288), (40, 288, 288))
```

The y here is the digit contained (or 0 for no digit) in each of the 81 cells for each sudoku image.

[16]:
```
y_train.shape, y_test.shape, y_train_all.shape, y_test_all.shape
```

[16]:
```
((160, 81), (40, 81), (160, 81), (40, 81))
```

[17]:
```
digit_train_img = []
digit_train_truth = []
digit_train_truth_all = []

for i, sudoku in tqdm(enumerate(X_train)):
                digit_train_img.extend(get_cells(sudoku, size=288))
                digit_train_truth.extend(y_train[i])
                digit_train_truth_all.extend(y_train_all[i])

digit_train_truth = np.array(digit_train_truth)
digit_train_img = np.array(digit_train_img)
digit_train_truth_all = np.array(digit_train_truth_all)
```

```
0it [00:00, ?it/s]
```

[18]:
```
print(digit_train_img.shape)
```

```
(12960, 32, 32)
```

[19]:
```
digit_train_img.shape, digit_train_truth.shape
```

[19]:
```
((12960, 32, 32), (12960,))
```

[20]:
```
digit_test_img = []
digit_test_truth = []
digit_test_truth_all = []

for i, sudoku in tqdm(enumerate(X_test)):
                digit_test_img.extend(get_cells(sudoku, size=288))
                digit_test_truth.extend(y_test[i])
                digit_test_truth_all.extend(y_test_all[i])

digit_test_truth = np.array(digit_test_truth)
digit_test_img = np.array(digit_test_img)
digit_test_truth_all = np.array(digit_test_truth_all)
```

```
0it [00:00, ?it/s]
```

[21]:
```
digit_test_img.shape, digit_test_truth.shape
```

[21]:
```
((3240, 32, 32), (3240,))
```

[23]:
```
digit_train_img = digit_train_img.reshape(-1, 1024)
digit_test_img = digit_test_img.reshape(-1, 1024)
```

[24]:
```
digit_train_img.shape
```

[24]:
```
(12960, 1024)
```

Finally, we obtain the dataset consisting of 12960 different 32x32 training images and the 12960 labels corresponding to them. Since our models take in a fixed-length vector of number, we flatten the 32x32 matrix into a 1024-element-long vector.

4 Using neural networks for this task

[25]:
```
# just to do one-hot encoding
from tensorflow.keras.utils import to_categorical
```

[26]:
```
training_data = list(zip(digit_train_img,␣
 ↪to_categorical(digit_train_truth_all)))
test_data = list(zip(digit_test_img, digit_test_truth_all))
```

4.1 Coding a neural network from scratch:

This is a simple implementation from neuralnetworksanddeeplearning.com (a great resource to learn NNs deeply) adapted to Python 3. Added code to track training accuracy as well.

[27]:
```
class Network(object):
    def __init__(self, sizes):
        self.num_layers = len(sizes)
        self.sizes = sizes
        self.biases = [np.random.randn(y) for y in sizes[1:]]
        self.weights = [np.random.randn(y, x)
                        for x, y in zip(sizes[:-1], sizes[1:])]

    def feedforward(self, a):
        for b, w in zip(self.biases, self.weights):
            a = sigmoid(np.dot(w, a)+b)
        return a

    def SGD(self, training_data, epochs, mini_batch_size, eta,
            test_data=None):
        if test_data: n_test = len(test_data)
        n = len(training_data)
        for j in range(epochs):
            random.shuffle(training_data)
            mini_batches = [
                training_data[k:k+mini_batch_size]
                for k in range(0, n, mini_batch_size)]
            for mini_batch in mini_batches:
```

```
                self.update_mini_batch(mini_batch, eta)
            if test_data:
                correct_test = self.evaluate(test_data)
                correct_train = self.evaluate_train(training_data)
                print("Epoch {0}: Val {1}/{2} - Accuracy: {3}, Train {4}/{5} -␣
↪Accuracy: {6}".format(
                    j, correct_test, n_test, round(correct_test / n_test, 3),
                    correct_train, len(training_data), round(correct_train /␣
↪len(training_data), 3)
                ))
            else:
                print("Epoch {0} complete".format(j))

    def update_mini_batch(self, mini_batch, eta):
        nabla_b = [np.zeros(b.shape) for b in self.biases]
        nabla_w = [np.zeros(w.shape) for w in self.weights]
        for x, y in mini_batch:
            delta_nabla_b, delta_nabla_w = self.backprop(x, y)
            nabla_b = [nb+dnb for nb, dnb in zip(nabla_b, delta_nabla_b)]
            nabla_w = [nw+dnw for nw, dnw in zip(nabla_w, delta_nabla_w)]
        self.weights = [w-(eta/len(mini_batch))*nw
                        for w, nw in zip(self.weights, nabla_w)]
        self.biases = [b-(eta/len(mini_batch))*nb
                       for b, nb in zip(self.biases, nabla_b)]

    def backprop(self, x, y):
        nabla_b = [np.zeros(b.shape) for b in self.biases]
        nabla_w = [np.zeros(w.shape) for w in self.weights]

        activation = x
        activations = [x]
        zs = []
        for b, w in zip(self.biases, self.weights):
            z = np.dot(w, activation)+b
            zs.append(z)
            activation = sigmoid(z)
            activations.append(activation)
        delta = self.cost_derivative(activations[-1], y) * \
            sigmoid_prime(zs[-1])
        nabla_b[-1] = delta
        nabla_w[-1] = np.dot(delta.reshape(-1, 1), activations[-2].reshape(-1,␣
↪1).transpose())
        for l in range(2, self.num_layers):
            z = zs[-l]
            sp = sigmoid_prime(z)
            delta = np.dot(self.weights[-l+1].transpose(), delta) * sp
            nabla_b[-l] = delta
```

```
            nabla_w[-l] = np.dot(delta.reshape(-1, 1), activations[-l-1].
↪reshape(-1, 1).transpose())
        return (nabla_b, nabla_w)

    def evaluate(self, test_data):
        test_results = [(np.argmax(self.feedforward(x)), y)
                        for (x, y) in test_data]
        return sum(int(x == y) for (x, y) in test_results)

    def evaluate_train(self, training_data):
        train_results = [(np.argmax(self.feedforward(x)), np.argmax(y))
                        for (x, y) in training_data]
        return sum(int(x == y) for (x, y) in train_results)

    def cost_derivative(self, output_activations, y):
        return (output_activations-y)

def sigmoid(z):
    return 1/(1 + np.exp(-z))

def sigmoid_prime(z):
    return sigmoid(z) * (1 - sigmoid(z))
```

Training a neural network with one hidden layer:

[39]:
```
net = Network([1024, 16, 10])
net.SGD(training_data, 10, 64, 2, test_data=test_data)
```

```
Epoch 0: Val 2105/3240 - Accuracy: 0.65, Train 8421/12960 - Accuracy: 0.65
Epoch 1: Val 2165/3240 - Accuracy: 0.668, Train 8622/12960 - Accuracy: 0.665
Epoch 2: Val 2199/3240 - Accuracy: 0.679, Train 8768/12960 - Accuracy: 0.677
Epoch 3: Val 2238/3240 - Accuracy: 0.691, Train 8940/12960 - Accuracy: 0.69
Epoch 4: Val 2251/3240 - Accuracy: 0.695, Train 9048/12960 - Accuracy: 0.698
Epoch 5: Val 2262/3240 - Accuracy: 0.698, Train 9115/12960 - Accuracy: 0.703
Epoch 6: Val 2280/3240 - Accuracy: 0.704, Train 9219/12960 - Accuracy: 0.711
Epoch 7: Val 2306/3240 - Accuracy: 0.712, Train 9296/12960 - Accuracy: 0.717
Epoch 8: Val 2313/3240 - Accuracy: 0.714, Train 9309/12960 - Accuracy: 0.718
Epoch 9: Val 2315/3240 - Accuracy: 0.715, Train 9344/12960 - Accuracy: 0.721
```

4.2 Using TensorFlow and Keras

While for a simple example like above, coding from scratch would work, but as models become more complex, a library like TensorFlow might be more suitable. Keras is a framework on top of TensorFlow that makes designing and testing neural networks even more simpler!

[29]:
```
from tensorflow.keras.layers import Dense, ReLU
from tensorflow.keras.models import Sequential
```

```
from tensorflow.keras.optimizers import Adam
```

4.2.1 Design a simple feed-forward neural network

Our model is a simple feed-forward neural network, where each layer takes input from the previous one and feeds the output to the next, so we use the Sequential model.

[30]:
```
model = Sequential()
```

[31]:
```
print(digit_train_img[0])
```

```
[0.27843137 0.27843137 0.3254902  … 0.04313725 0.21960784 0.60784314]
```

1. To add a fully-connected hidden layer to our model, we can use the **Dense** layer.
2. Since this is the first (hidden) layer of our model, we also mention the input shape (`input_shape`) - the dimensions of our input layer - in our case this is 1024 (the 32 x 32 image we have).
3. Let's use a hidden layer of size 16, and then add an activation function. In Keras, Activation functions also work like Layers. Here we use the ReLU activation, sigmoid can be used all.
4. Finally, our output layer has 10 neurons as we have 10 classes (1-9 being digits, 0 being no digit). We apply an activation function through the activation parameter rather than a separate layer. Both these ways of applying activations functions are acceptable and equivalent.

[32]:
```
model.add(Dense(16, input_shape=(digit_train_img.shape[1], )))
model.add(ReLU()) # Can be sigmoid too
model.add(Dense(10, activation='sigmoid'))
```

Then we compile our model, mentioning what loss and optimizer we want to use along with any metrics we want to track.

Since this is a classification problem, accuracy can be one metric to track.

[33]:
```
model.compile(loss='sparse_categorical_crossentropy',
              optimizer=Adam(learning_rate=0.01),
              metrics=['accuracy'])
```

[34]:
```
model.fit(digit_train_img,
          digit_train_truth_all,
          validation_data=(digit_test_img, digit_test_truth_all),
          batch_size=128,
          epochs=20)
```

```
Epoch 1/20
102/102 [==============================] - 1s 5ms/step - loss: 1.0230 -
accuracy: 0.6678 - val_loss: 0.8524 - val_accuracy: 0.7046
Epoch 2/20
102/102 [==============================] - 0s 3ms/step - loss: 0.8681 -
```

```
accuracy: 0.7231 - val_loss: 0.7823 - val_accuracy: 0.7488
Epoch 3/20
102/102 [==============================] - 0s 3ms/step - loss: 0.8011 -
accuracy: 0.7527 - val_loss: 0.7381 - val_accuracy: 0.7716
Epoch 4/20
102/102 [==============================] - 0s 3ms/step - loss: 0.7625 -
accuracy: 0.7679 - val_loss: 0.7110 - val_accuracy: 0.7870
Epoch 5/20
102/102 [==============================] - 0s 3ms/step - loss: 0.7233 -
accuracy: 0.7790 - val_loss: 0.7155 - val_accuracy: 0.7917
Epoch 6/20
102/102 [==============================] - 0s 3ms/step - loss: 0.6931 -
accuracy: 0.7886 - val_loss: 0.6770 - val_accuracy: 0.7941
Epoch 7/20
102/102 [==============================] - 0s 3ms/step - loss: 0.6724 -
accuracy: 0.7972 - val_loss: 0.6630 - val_accuracy: 0.7948
Epoch 8/20
102/102 [==============================] - 0s 3ms/step - loss: 0.6573 -
accuracy: 0.7961 - val_loss: 0.6787 - val_accuracy: 0.7923
Epoch 9/20
102/102 [==============================] - 0s 3ms/step - loss: 0.6597 -
accuracy: 0.7975 - val_loss: 0.6831 - val_accuracy: 0.7981
Epoch 10/20
102/102 [==============================] - 0s 3ms/step - loss: 0.6451 -
accuracy: 0.8007 - val_loss: 0.6681 - val_accuracy: 0.8043
Epoch 11/20
102/102 [==============================] - 0s 3ms/step - loss: 0.6317 -
accuracy: 0.8032 - val_loss: 0.6694 - val_accuracy: 0.7957
Epoch 12/20
102/102 [==============================] - 0s 3ms/step - loss: 0.6174 -
accuracy: 0.8098 - val_loss: 0.6542 - val_accuracy: 0.8130
Epoch 13/20
102/102 [==============================] - 0s 3ms/step - loss: 0.6093 -
accuracy: 0.8113 - val_loss: 0.6857 - val_accuracy: 0.7889
Epoch 14/20
102/102 [==============================] - 0s 3ms/step - loss: 0.6070 -
accuracy: 0.8127 - val_loss: 0.7012 - val_accuracy: 0.7889
Epoch 15/20
102/102 [==============================] - 0s 3ms/step - loss: 0.6001 -
accuracy: 0.8140 - val_loss: 0.6604 - val_accuracy: 0.8093
Epoch 16/20
102/102 [==============================] - 0s 3ms/step - loss: 0.6070 -
accuracy: 0.8113 - val_loss: 0.6673 - val_accuracy: 0.8052
Epoch 17/20
102/102 [==============================] - 0s 3ms/step - loss: 0.5953 -
accuracy: 0.8159 - val_loss: 0.6710 - val_accuracy: 0.8040
Epoch 18/20
102/102 [==============================] - 0s 3ms/step - loss: 0.5880 -
```

```
accuracy: 0.8177 - val_loss: 0.6611 - val_accuracy: 0.8031
Epoch 19/20
102/102 [==============================] - 0s 3ms/step - loss: 0.5735 -
accuracy: 0.8230 - val_loss: 0.6630 - val_accuracy: 0.8062
Epoch 20/20
102/102 [==============================] - 0s 3ms/step - loss: 0.5827 -
accuracy: 0.8189 - val_loss: 0.6667 - val_accuracy: 0.8031
```

[34]:
```
<keras.callbacks.History at 0x7fd736f56b50>
```

[35]:
```
def get_weights(model, i):
    ''' Helper function to retrieve a hidden layer nodes weight mapping to the␣
 ↪input layer '''
    return model.layers[0].get_weights()[0][:, i].reshape(32, 32)
```

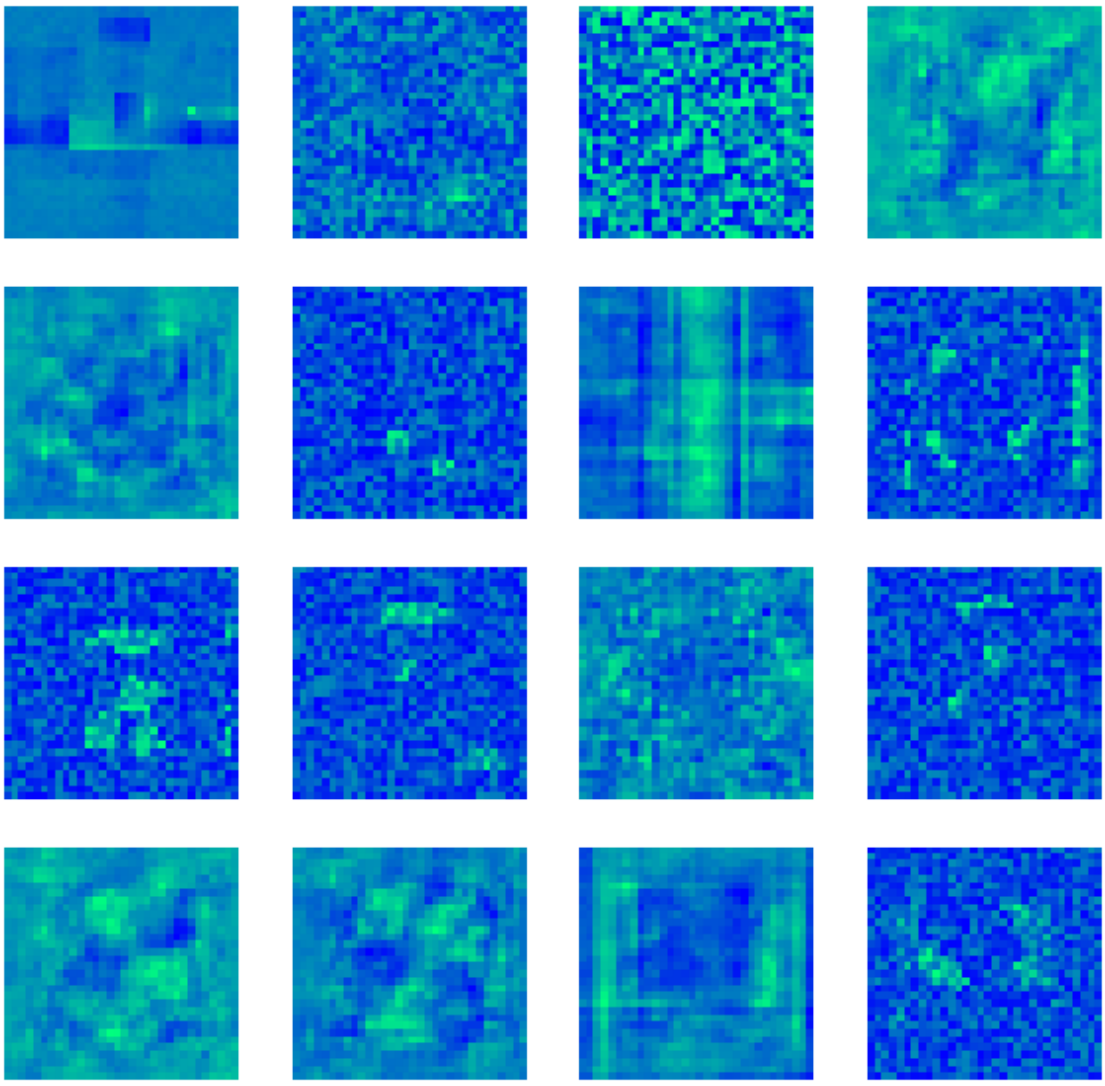
[36]:
```
model.save('sudoku_solver_nn.h5')
```

4.3 Visualizing the trained weights

After training our network, taking the first layer's weights (a 1024 * 16) matrix, we can find out what kind of patterns activate/trigger each of the 16 hidden layer nodes.

This can be done by reshaping the 1024 x 1 weight vector for each hidden layer node into a 32 x 32 matrix and plotting it.

[37]:
```
ax = plt.figure(figsize=(10, 10))
for i in range(16):
    plt.subplot(4, 4, i+1)
    plt.axis('off')
    plt.imshow(get_weights(model, i), cmap='winter')
```



- It's pretty clear that the model learns different lines and curves involved in classifying the digits (as well as the lack of any digit!)
- What makes this interesting is the fact that we **never explicitly mentioned** or conveyed the neural network to learn these features, it learned it by itself. All we ever did was to program a method to reduce the number of mistakes the model makes, this drives/forces the network to learn these patterns in order to achieve that.
- By restricting the number of hidden layer nodes to 16, we might not achieve the best performance but it gives us insight into what **patterns** a neural network learns to recognise when it's constrained.

[42]:
```
from google.colab import drive
drive.mount('/content/drive')
!ls
```

```
# !jupyter nbconvert --to PDF "Untitled.ipynb"
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
Data         img_282.png  img_466.png  img_638.png  img_849.png
drive        img_284.png  img_467.png  img_639.png  img_852.png
img_0.png    img_287.png  img_469.png  img_644.png  img_859.png
img_100.png  img_28.png   img_470.png  img_647.png  img_861.png
img_101.png  img_292.png  img_472.png  img_648.png  img_862.png
img_107.png  img_295.png  img_475.png  img_651.png  img_863.png
img_108.png  img_298.png  img_477.png  img_652.png  img_864.png
img_10.png   img_29.png   img_479.png  img_657.png  img_865.png
img_111.png  img_308.png  img_480.png  img_659.png  img_866.png
img_113.png  img_309.png  img_491.png  img_661.png  img_867.png
img_114.png  img_30.png   img_495.png  img_665.png  img_868.png
img_115.png  img_311.png  img_499.png  img_666.png  img_86.png
img_118.png  img_313.png  img_49.png   img_667.png  img_870.png
img_119.png  img_315.png  img_500.png  img_671.png  img_871.png
img_11.png   img_317.png  img_502.png  img_677.png  img_873.png
img_120.png  img_318.png  img_503.png  img_678.png  img_874.png
img_12.png   img_321.png  img_505.png  img_686.png  img_876.png
img_133.png  img_326.png  img_506.png  img_690.png  img_877.png
img_138.png  img_327.png  img_512.png  img_698.png  img_878.png
img_139.png  img_32.png   img_513.png  img_699.png  img_879.png
img_140.png  img_331.png  img_516.png  img_69.png   img_881.png
img_144.png  img_332.png  img_518.png  img_705.png  img_892.png
img_145.png  img_333.png  img_519.png  img_709.png  img_893.png
img_146.png  img_336.png  img_520.png  img_710.png  img_895.png
img_149.png  img_343.png  img_523.png  img_711.png  img_896.png
img_14.png   img_34.png   img_524.png  img_715.png  img_901.png
img_151.png  img_353.png  img_525.png  img_717.png  img_903.png
img_152.png  img_355.png  img_538.png  img_719.png  img_906.png
img_159.png  img_358.png  img_543.png  img_71.png   img_908.png
img_160.png  img_361.png  img_544.png  img_724.png  img_90.png
img_161.png  img_362.png  img_545.png  img_725.png  img_910.png
img_162.png  img_363.png  img_549.png  img_728.png  img_912.png
img_16.png   img_365.png  img_54.png   img_730.png  img_913.png
img_172.png  img_366.png  img_550.png  img_736.png  img_917.png
img_173.png  img_367.png  img_551.png  img_738.png  img_918.png
img_174.png  img_36.png   img_554.png  img_73.png   img_924.png
img_176.png  img_370.png  img_556.png  img_740.png  img_926.png
img_178.png  img_373.png  img_557.png  img_743.png  img_929.png
img_184.png  img_375.png  img_564.png  img_746.png  img_930.png
img_185.png  img_37.png   img_565.png  img_748.png  img_932.png
img_188.png  img_385.png  img_566.png  img_750.png  img_933.png
img_190.png  img_392.png  img_567.png  img_751.png  img_936.png
img_191.png  img_395.png  img_568.png  img_758.png  img_938.png
```

```
img_192.png  img_396.png  img_571.png  img_763.png  img_944.png
img_194.png  img_397.png  img_575.png  img_767.png  img_945.png
img_196.png  img_401.png  img_581.png  img_769.png  img_949.png
img_198.png  img_402.png  img_582.png  img_76.png   img_94.png
img_199.png  img_405.png  img_583.png  img_771.png  img_950.png
img_204.png  img_406.png  img_584.png  img_775.png  img_952.png
img_211.png  img_407.png  img_589.png  img_780.png  img_954.png
img_216.png  img_408.png  img_591.png  img_787.png  img_956.png
img_221.png  img_409.png  img_592.png  img_788.png  img_959.png
img_224.png  img_414.png  img_596.png  img_78.png   img_95.png
img_22.png   img_416.png  img_597.png  img_790.png  img_961.png
img_231.png  img_417.png  img_59.png   img_792.png  img_966.png
img_233.png  img_419.png  img_600.png  img_795.png  img_967.png
img_235.png  img_420.png  img_601.png  img_798.png  img_969.png
img_238.png  img_422.png  img_604.png  img_800.png  img_970.png
img_23.png   img_423.png  img_609.png  img_802.png  img_977.png
img_240.png  img_428.png  img_611.png  img_808.png  img_97.png
img_245.png  img_42.png   img_615.png  img_810.png  img_981.png
img_248.png  img_431.png  img_617.png  img_812.png  img_985.png
img_249.png  img_432.png  img_621.png  img_817.png  img_986.png
img_251.png  img_435.png  img_622.png  img_819.png  img_988.png
img_253.png  img_437.png  img_625.png  img_823.png  img_989.png
img_255.png  img_438.png  img_626.png  img_827.png  img_98.png
img_257.png  img_441.png  img_627.png  img_831.png  img_991.png
img_258.png  img_442.png  img_629.png  img_832.png  img_992.png
img_268.png  img_443.png  img_62.png   img_834.png  img_998.png
img_26.png   img_449.png  img_631.png  img_837.png  img_999.png
img_271.png  img_451.png  img_632.png  img_839.png  sample_data
img_274.png  img_452.png  img_636.png  img_842.png  sudoku_solver_nn.h5
img_279.png  img_456.png  img_637.png  img_848.png
```

[46]:
```
import os
os.chdir("drive/MyDrive/Colab Notebooks")
```

[ ]:
```
!apt-get install texlive texlive-xetex texlive-latex-extra pandoc
!pip install pypandoc
!jupyter nbconvert --to PDF "SudokuSolver_P2.ipynb"
```

```
Reading package lists… Done
Building dependency tree
Reading state information… Done
pandoc is already the newest version (1.19.2.4~dfsg-1build4).
pandoc set to manually installed.
The following additional packages will be installed:
  fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-texgyre
  javascript-common libcupsfilters1 libcupsimage2 libgs9 libgs9-common
  libijs-0.35 libjbig2dec0 libjs-jquery libkpathsea6 libpotrace0 libptexenc1
```

```
  libruby2.5 libsynctex1 libtexlua52 libtexluajit2 libzzip-0-13 lmodern
  poppler-data preview-latex-style rake ruby ruby-did-you-mean ruby-minitest
  ruby-net-telnet ruby-power-assert ruby-test-unit ruby2.5
  rubygems-integration t1utils tex-common tex-gyre texlive-base
  texlive-binaries texlive-fonts-recommended texlive-latex-base
  texlive-latex-recommended texlive-pictures texlive-plain-generic tipa
Suggested packages:
  fonts-noto apache2 | lighttpd | httpd poppler-utils ghostscript
  fonts-japanese-mincho | fonts-ipafont-mincho fonts-japanese-gothic
  | fonts-ipafont-gothic fonts-arphic-ukai fonts-arphic-uming fonts-nanum ri
  ruby-dev bundler debhelper gv | postscript-viewer perl-tk xpdf-reader
  | pdf-viewer texlive-fonts-recommended-doc texlive-latex-base-doc
  python-pygments icc-profiles libfile-which-perl
  libspreadsheet-parseexcel-perl texlive-latex-extra-doc
  texlive-latex-recommended-doc texlive-pstricks dot2tex prerex ruby-tcltk
  | libtcltk-ruby texlive-pictures-doc vprerex
The following NEW packages will be installed:
  fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-texgyre
  javascript-common libcupsfilters1 libcupsimage2 libgs9 libgs9-common
  libijs-0.35 libjbig2dec0 libjs-jquery libkpathsea6 libpotrace0 libptexenc1
  libruby2.5 libsynctex1 libtexlua52 libtexluajit2 libzzip-0-13 lmodern
  poppler-data preview-latex-style rake ruby ruby-did-you-mean ruby-minitest
  ruby-net-telnet ruby-power-assert ruby-test-unit ruby2.5
  rubygems-integration t1utils tex-common tex-gyre texlive texlive-base
  texlive-binaries texlive-fonts-recommended texlive-latex-base
  texlive-latex-extra texlive-latex-recommended texlive-pictures
  texlive-plain-generic texlive-xetex tipa
0 upgraded, 47 newly installed, 0 to remove and 39 not upgraded.
Need to get 146 MB of archives.
After this operation, 460 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-droid-fallback
all 1:6.0.1r16-1.1 [1,805 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-lato all 2.0-2
[2,698 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic/main amd64 poppler-data all
0.4.8-2 [1,479 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic/main amd64 tex-common all 6.09
[33.0 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-lmodern all
2.004.5-3 [4,551 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-noto-mono all
20171026-2 [75.5 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic/universe amd64 fonts-texgyre all
20160520-1 [8,761 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic/main amd64 javascript-common all
11 [6,066 B]
Get:9 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libcupsfilters1
amd64 1.20.2-0ubuntu3.1 [108 kB]
```

```
Get:10 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libcupsimage2
amd64 2.2.7-1ubuntu2.8 [18.6 kB]
Get:11 http://archive.ubuntu.com/ubuntu bionic/main amd64 libijs-0.35 amd64
0.35-13 [15.5 kB]
Get:12 http://archive.ubuntu.com/ubuntu bionic/main amd64 libjbig2dec0 amd64
0.13-6 [55.9 kB]
Get:13 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libgs9-common
all 9.26~dfsg+0-0ubuntu0.18.04.15 [5,092 kB]
Get:14 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libgs9 amd64
9.26~dfsg+0-0ubuntu0.18.04.15 [2,265 kB]
Get:15 http://archive.ubuntu.com/ubuntu bionic/main amd64 libjs-jquery all
3.2.1-1 [152 kB]
Get:16 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libkpathsea6
amd64 2017.20170613.44572-8ubuntu0.1 [54.9 kB]
Get:17 http://archive.ubuntu.com/ubuntu bionic/main amd64 libpotrace0 amd64
1.14-2 [17.4 kB]
Get:18 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libptexenc1
amd64 2017.20170613.44572-8ubuntu0.1 [34.5 kB]
Get:19 http://archive.ubuntu.com/ubuntu bionic/main amd64 rubygems-integration
all 1.11 [4,994 B]
Get:20 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 ruby2.5 amd64
2.5.1-1ubuntu1.11 [48.6 kB]
Get:21 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby amd64 1:2.5.1
[5,712 B]
Get:22 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 rake all
12.3.1-1ubuntu0.1 [44.9 kB]
Get:23 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-did-you-mean all
1.2.0-2 [9,700 B]
Get:24 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-minitest all
5.10.3-1 [38.6 kB]
Get:25 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-net-telnet all
0.1.1-2 [12.6 kB]
Get:26 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-power-assert all
0.3.0-1 [7,952 B]
Get:27 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-test-unit all
3.2.5-1 [61.1 kB]
Get:28 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libruby2.5
amd64 2.5.1-1ubuntu1.11 [3,072 kB]
Get:29 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libsynctex1
amd64 2017.20170613.44572-8ubuntu0.1 [41.4 kB]
Get:30 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libtexlua52
amd64 2017.20170613.44572-8ubuntu0.1 [91.2 kB]
Get:31 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libtexluajit2
amd64 2017.20170613.44572-8ubuntu0.1 [230 kB]
Get:32 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libzzip-0-13
amd64 0.13.62-3.1ubuntu0.18.04.1 [26.0 kB]
Get:33 http://archive.ubuntu.com/ubuntu bionic/main amd64 lmodern all 2.004.5-3
[9,631 kB]
```

```
Get:34 http://archive.ubuntu.com/ubuntu bionic/main amd64 preview-latex-style
all 11.91-1ubuntu1 [185 kB]
Get:35 http://archive.ubuntu.com/ubuntu bionic/main amd64 t1utils amd64 1.41-2
[56.0 kB]
Get:36 http://archive.ubuntu.com/ubuntu bionic/universe amd64 tex-gyre all
20160520-1 [4,998 kB]
Get:37 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 texlive-
binaries amd64 2017.20170613.44572-8ubuntu0.1 [8,179 kB]
Get:38 http://archive.ubuntu.com/ubuntu bionic/main amd64 texlive-base all
2017.20180305-1 [18.7 MB]
Get:39 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-fonts-
recommended all 2017.20180305-1 [5,262 kB]
Get:40 http://archive.ubuntu.com/ubuntu bionic/main amd64 texlive-latex-base all
2017.20180305-1 [951 kB]
Get:41 http://archive.ubuntu.com/ubuntu bionic/main amd64 texlive-latex-
recommended all 2017.20180305-1 [14.9 MB]
Get:42 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive all
2017.20180305-1 [14.4 kB]
Get:43 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-pictures
all 2017.20180305-1 [4,026 kB]
Get:44 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-latex-
extra all 2017.20180305-2 [10.6 MB]
Get:45 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-plain-
generic all 2017.20180305-2 [23.6 MB]
Get:46 http://archive.ubuntu.com/ubuntu bionic/universe amd64 tipa all 2:1.3-20
[2,978 kB]
Get:47 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-xetex all
2017.20180305-1 [10.7 MB]
Fetched 146 MB in 2s (65.2 MB/s)
Extracting templates from packages: 100%
Preconfiguring packages …
Selecting previously unselected package fonts-droid-fallback.
(Reading database … 156210 files and directories currently installed.)
Preparing to unpack …/00-fonts-droid-fallback_1%3a6.0.1r16-1.1_all.deb …
Unpacking fonts-droid-fallback (1:6.0.1r16-1.1) …
Selecting previously unselected package fonts-lato.
Preparing to unpack …/01-fonts-lato_2.0-2_all.deb …
Unpacking fonts-lato (2.0-2) …
Selecting previously unselected package poppler-data.
Preparing to unpack …/02-poppler-data_0.4.8-2_all.deb …
Unpacking poppler-data (0.4.8-2) …
Selecting previously unselected package tex-common.
Preparing to unpack …/03-tex-common_6.09_all.deb …
Unpacking tex-common (6.09) …
Selecting previously unselected package fonts-lmodern.
Preparing to unpack …/04-fonts-lmodern_2.004.5-3_all.deb …
Unpacking fonts-lmodern (2.004.5-3) …
Selecting previously unselected package fonts-noto-mono.
```

```
Preparing to unpack …/05-fonts-noto-mono_20171026-2_all.deb …
Unpacking fonts-noto-mono (20171026-2) …
Selecting previously unselected package fonts-texgyre.
Preparing to unpack …/06-fonts-texgyre_20160520-1_all.deb …
Unpacking fonts-texgyre (20160520-1) …
Selecting previously unselected package javascript-common.
Preparing to unpack …/07-javascript-common_11_all.deb …
Unpacking javascript-common (11) …
Selecting previously unselected package libcupsfilters1:amd64.
Preparing to unpack …/08-libcupsfilters1_1.20.2-0ubuntu3.1_amd64.deb …
Unpacking libcupsfilters1:amd64 (1.20.2-0ubuntu3.1) …
Selecting previously unselected package libcupsimage2:amd64.
Preparing to unpack …/09-libcupsimage2_2.2.7-1ubuntu2.8_amd64.deb …
Unpacking libcupsimage2:amd64 (2.2.7-1ubuntu2.8) …
Selecting previously unselected package libijs-0.35:amd64.
Preparing to unpack …/10-libijs-0.35_0.35-13_amd64.deb …
Unpacking libijs-0.35:amd64 (0.35-13) …
Selecting previously unselected package libjbig2dec0:amd64.
Preparing to unpack …/11-libjbig2dec0_0.13-6_amd64.deb …
Unpacking libjbig2dec0:amd64 (0.13-6) …
Selecting previously unselected package libgs9-common.
Preparing to unpack …/12-libgs9-common_9.26~dfsg+0-0ubuntu0.18.04.15_all.deb
…
Unpacking libgs9-common (9.26~dfsg+0-0ubuntu0.18.04.15) …
Selecting previously unselected package libgs9:amd64.
Preparing to unpack …/13-libgs9_9.26~dfsg+0-0ubuntu0.18.04.15_amd64.deb …
Unpacking libgs9:amd64 (9.26~dfsg+0-0ubuntu0.18.04.15) …
Selecting previously unselected package libjs-jquery.
Preparing to unpack …/14-libjs-jquery_3.2.1-1_all.deb …
Unpacking libjs-jquery (3.2.1-1) …
Selecting previously unselected package libkpathsea6:amd64.
Preparing to unpack …/15-libkpathsea6_2017.20170613.44572-8ubuntu0.1_amd64.deb
…
Unpacking libkpathsea6:amd64 (2017.20170613.44572-8ubuntu0.1) …
Selecting previously unselected package libpotrace0.
Preparing to unpack …/16-libpotrace0_1.14-2_amd64.deb …
Unpacking libpotrace0 (1.14-2) …
Selecting previously unselected package libptexenc1:amd64.
Preparing to unpack …/17-libptexenc1_2017.20170613.44572-8ubuntu0.1_amd64.deb
…
Unpacking libptexenc1:amd64 (2017.20170613.44572-8ubuntu0.1) …
Selecting previously unselected package rubygems-integration.
Preparing to unpack …/18-rubygems-integration_1.11_all.deb …
Unpacking rubygems-integration (1.11) …
Selecting previously unselected package ruby2.5.
Preparing to unpack …/19-ruby2.5_2.5.1-1ubuntu1.11_amd64.deb …
Unpacking ruby2.5 (2.5.1-1ubuntu1.11) …
Selecting previously unselected package ruby.
```

```
Preparing to unpack …/20-ruby_1%3a2.5.1_amd64.deb …
Unpacking ruby (1:2.5.1) …
Selecting previously unselected package rake.
Preparing to unpack …/21-rake_12.3.1-1ubuntu0.1_all.deb …
Unpacking rake (12.3.1-1ubuntu0.1) …
Selecting previously unselected package ruby-did-you-mean.
Preparing to unpack …/22-ruby-did-you-mean_1.2.0-2_all.deb …
Unpacking ruby-did-you-mean (1.2.0-2) …
Selecting previously unselected package ruby-minitest.
Preparing to unpack …/23-ruby-minitest_5.10.3-1_all.deb …
Unpacking ruby-minitest (5.10.3-1) …
Selecting previously unselected package ruby-net-telnet.
Preparing to unpack …/24-ruby-net-telnet_0.1.1-2_all.deb …
Unpacking ruby-net-telnet (0.1.1-2) …
Selecting previously unselected package ruby-power-assert.
Preparing to unpack …/25-ruby-power-assert_0.3.0-1_all.deb …
Unpacking ruby-power-assert (0.3.0-1) …
Selecting previously unselected package ruby-test-unit.
Preparing to unpack …/26-ruby-test-unit_3.2.5-1_all.deb …
Unpacking ruby-test-unit (3.2.5-1) …
Selecting previously unselected package libruby2.5:amd64.
Preparing to unpack …/27-libruby2.5_2.5.1-1ubuntu1.11_amd64.deb …
Unpacking libruby2.5:amd64 (2.5.1-1ubuntu1.11) …
Selecting previously unselected package libsynctex1:amd64.
Preparing to unpack …/28-libsynctex1_2017.20170613.44572-8ubuntu0.1_amd64.deb
…
Unpacking libsynctex1:amd64 (2017.20170613.44572-8ubuntu0.1) …
Selecting previously unselected package libtexlua52:amd64.
Preparing to unpack …/29-libtexlua52_2017.20170613.44572-8ubuntu0.1_amd64.deb
…
Unpacking libtexlua52:amd64 (2017.20170613.44572-8ubuntu0.1) …
Selecting previously unselected package libtexluajit2:amd64.
Preparing to unpack
…/30-libtexluajit2_2017.20170613.44572-8ubuntu0.1_amd64.deb …
Unpacking libtexluajit2:amd64 (2017.20170613.44572-8ubuntu0.1) …
Selecting previously unselected package libzzip-0-13:amd64.
Preparing to unpack …/31-libzzip-0-13_0.13.62-3.1ubuntu0.18.04.1_amd64.deb …
Unpacking libzzip-0-13:amd64 (0.13.62-3.1ubuntu0.18.04.1) …
Selecting previously unselected package lmodern.
Preparing to unpack …/32-lmodern_2.004.5-3_all.deb …
Unpacking lmodern (2.004.5-3) …
Selecting previously unselected package preview-latex-style.
Preparing to unpack …/33-preview-latex-style_11.91-1ubuntu1_all.deb …
Unpacking preview-latex-style (11.91-1ubuntu1) …
Selecting previously unselected package t1utils.
Preparing to unpack …/34-t1utils_1.41-2_amd64.deb …
Unpacking t1utils (1.41-2) …
Selecting previously unselected package tex-gyre.
```

```
Preparing to unpack …/35-tex-gyre_20160520-1_all.deb …
Unpacking tex-gyre (20160520-1) …
Selecting previously unselected package texlive-binaries.
Preparing to unpack …/36-texlive-
binaries_2017.20170613.44572-8ubuntu0.1_amd64.deb …
Unpacking texlive-binaries (2017.20170613.44572-8ubuntu0.1) …
Selecting previously unselected package texlive-base.
Preparing to unpack …/37-texlive-base_2017.20180305-1_all.deb …
Unpacking texlive-base (2017.20180305-1) …
Selecting previously unselected package texlive-fonts-recommended.
Preparing to unpack …/38-texlive-fonts-recommended_2017.20180305-1_all.deb …
Unpacking texlive-fonts-recommended (2017.20180305-1) …
Selecting previously unselected package texlive-latex-base.
Preparing to unpack …/39-texlive-latex-base_2017.20180305-1_all.deb …
Unpacking texlive-latex-base (2017.20180305-1) …
Selecting previously unselected package texlive-latex-recommended.
Preparing to unpack …/40-texlive-latex-recommended_2017.20180305-1_all.deb …
Unpacking texlive-latex-recommended (2017.20180305-1) …
Selecting previously unselected package texlive.
Preparing to unpack …/41-texlive_2017.20180305-1_all.deb …
Unpacking texlive (2017.20180305-1) …
Selecting previously unselected package texlive-pictures.
Preparing to unpack …/42-texlive-pictures_2017.20180305-1_all.deb …
Unpacking texlive-pictures (2017.20180305-1) …
Selecting previously unselected package texlive-latex-extra.
Preparing to unpack …/43-texlive-latex-extra_2017.20180305-2_all.deb …
Unpacking texlive-latex-extra (2017.20180305-2) …
Selecting previously unselected package texlive-plain-generic.
Preparing to unpack …/44-texlive-plain-generic_2017.20180305-2_all.deb …
Unpacking texlive-plain-generic (2017.20180305-2) …
Selecting previously unselected package tipa.
Preparing to unpack …/45-tipa_2%3a1.3-20_all.deb …
Unpacking tipa (2:1.3-20) …
Selecting previously unselected package texlive-xetex.
Preparing to unpack …/46-texlive-xetex_2017.20180305-1_all.deb …
Unpacking texlive-xetex (2017.20180305-1) …
Setting up libgs9-common (9.26~dfsg+0-0ubuntu0.18.04.15) …
Setting up libkpathsea6:amd64 (2017.20170613.44572-8ubuntu0.1) …
Setting up libjs-jquery (3.2.1-1) …
Setting up libtexlua52:amd64 (2017.20170613.44572-8ubuntu0.1) …
Setting up fonts-droid-fallback (1:6.0.1r16-1.1) …
Setting up libsynctex1:amd64 (2017.20170613.44572-8ubuntu0.1) …
Setting up libptexenc1:amd64 (2017.20170613.44572-8ubuntu0.1) …
Setting up tex-common (6.09) …
update-language: texlive-base not installed and configured, doing nothing!
Setting up poppler-data (0.4.8-2) …
Setting up tex-gyre (20160520-1) …
Setting up preview-latex-style (11.91-1ubuntu1) …
```

```
Setting up fonts-texgyre (20160520-1) …
Setting up fonts-noto-mono (20171026-2) …
Setting up fonts-lato (2.0-2) …
Setting up libcupsfilters1:amd64 (1.20.2-0ubuntu3.1) …
Setting up libcupsimage2:amd64 (2.2.7-1ubuntu2.8) …
Setting up libjbig2dec0:amd64 (0.13-6) …
Setting up ruby-did-you-mean (1.2.0-2) …
Setting up t1utils (1.41-2) …
Setting up ruby-net-telnet (0.1.1-2) …
Setting up libijs-0.35:amd64 (0.35-13) …
Setting up rubygems-integration (1.11) …
Setting up libpotrace0 (1.14-2) …
Setting up javascript-common (11) …
Setting up ruby-minitest (5.10.3-1) …
Setting up libzzip-0-13:amd64 (0.13.62-3.1ubuntu0.18.04.1) …
Setting up libgs9:amd64 (9.26~dfsg+0-0ubuntu0.18.04.15) …
Setting up libtexluajit2:amd64 (2017.20170613.44572-8ubuntu0.1) …
Setting up fonts-lmodern (2.004.5-3) …
Setting up ruby-power-assert (0.3.0-1) …
Setting up texlive-binaries (2017.20170613.44572-8ubuntu0.1) …
update-alternatives: using /usr/bin/xdvi-xaw to provide /usr/bin/xdvi.bin
(xdvi.bin) in auto mode
update-alternatives: using /usr/bin/bibtex.original to provide /usr/bin/bibtex
(bibtex) in auto mode
Setting up texlive-base (2017.20180305-1) …
mktexlsr: Updating /var/lib/texmf/ls-R-TEXLIVEDIST…
mktexlsr: Updating /var/lib/texmf/ls-R-TEXMFMAIN…
mktexlsr: Updating /var/lib/texmf/ls-R…
mktexlsr: Done.
tl-paper: setting paper size for dvips to a4:
/var/lib/texmf/dvips/config/config-paper.ps
tl-paper: setting paper size for dvipdfmx to a4:
/var/lib/texmf/dvipdfmx/dvipdfmx-paper.cfg
tl-paper: setting paper size for xdvi to a4: /var/lib/texmf/xdvi/XDvi-paper
tl-paper: setting paper size for pdftex to a4:
/var/lib/texmf/tex/generic/config/pdftexconfig.tex
Setting up texlive-fonts-recommended (2017.20180305-1) …
Setting up texlive-plain-generic (2017.20180305-2) …
Setting up texlive-latex-base (2017.20180305-1) …
Setting up lmodern (2.004.5-3) …
Setting up texlive-latex-recommended (2017.20180305-1) …
Setting up texlive-pictures (2017.20180305-1) …
Setting up tipa (2:1.3-20) …
Regenerating '/var/lib/texmf/fmtutil.cnf-DEBIAN'… done.
Regenerating '/var/lib/texmf/fmtutil.cnf-TEXLIVEDIST'… done.
update-fmtutil has updated the following file(s):
        /var/lib/texmf/fmtutil.cnf-DEBIAN
        /var/lib/texmf/fmtutil.cnf-TEXLIVEDIST
```

```
If you want to activate the changes in the above file(s),
you should run fmtutil-sys or fmtutil.
Setting up texlive (2017.20180305-1) …
Setting up texlive-latex-extra (2017.20180305-2) …
Setting up texlive-xetex (2017.20180305-1) …
Setting up ruby2.5 (2.5.1-1ubuntu1.11) …
Setting up ruby (1:2.5.1) …
Setting up ruby-test-unit (3.2.5-1) …
Setting up rake (12.3.1-1ubuntu0.1) …
Setting up libruby2.5:amd64 (2.5.1-1ubuntu1.11) …
Processing triggers for mime-support (3.60ubuntu1) …
Processing triggers for libc-bin (2.27-3ubuntu1.3) …
/sbin/ldconfig.real: /usr/local/lib/python3.7/dist-
packages/ideep4py/lib/libmkldnn.so.0 is not a symbolic link

Processing triggers for man-db (2.8.3-2ubuntu0.1) …
Processing triggers for fontconfig (2.12.6-0ubuntu2) …
Processing triggers for tex-common (6.09) …
Running updmap-sys. This may take some time…
```