

```
In [1]: import numpy as np
def reweight_distribution(original_distribution, temperature=0.5):
    distribution = np.log(original_distribution) / temperature
    distribution = np.exp(distribution)
    return distribution / np.sum(distribution)
```

```
In [2]: import keras
import numpy as np
path = keras.utils.get_file(
    'nietzsche.txt',
    origin='https://s3.amazonaws.com/text-datasets/nietzsche.txt')
text = open(path).read().lower()
print('Corpus length:', len(text))
```

Corpus length: 600901

```
In [3]: maxlen = 60
step = 3
sentences = []
next_chars = []
for i in range(0, len(text) - maxlen, step):
    sentences.append(text[i: i + maxlen])
    next_chars.append(text[i + maxlen])
print('Number of sequences:', len(sentences))
chars = sorted(list(set(text)))
print('Unique characters:', len(chars))
char_indices = dict((char, chars.index(char)) for char in chars)
print('Vectorization...')
x = np.zeros((len(sentences), maxlen, len(chars)), dtype=np.bool)
y = np.zeros((len(sentences), len(chars)), dtype=np.bool)
for i, sentence in enumerate(sentences):
    for t, char in enumerate(sentence):
        x[i, t, char_indices[char]] = 1
    y[i, char_indices[next_chars[i]]] = 1
```

Number of sequences: 200281

Unique characters: 59

Vectorization...

C:\Users\theoj\AppData\Local\Temp\ipykernel_25400\321955488.py:13: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool`. To silence this warning, use `bool` by itself. Doing this will not modify any behavior and is safe. If you specifically wanted the numpy scalar type, use `np.bool_` here.

Deprecated in NumPy 1.20; for more details and guidance: <https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations>

```
x = np.zeros((len(sentences), maxlen, len(chars)), dtype=np.bool)
```

C:\Users\theoj\AppData\Local\Temp\ipykernel_25400\321955488.py:14: DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool`. To silence this warning, use `bool` by itself. Doing this will not modify any behavior and is safe. If you specifically wanted the numpy scalar type, use `np.bool_` here.

Deprecated in NumPy 1.20; for more details and guidance: <https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations>

```
y = np.zeros((len(sentences), len(chars)), dtype=np.bool)
```

```
In [4]: from keras import layers
model = keras.models.Sequential()
model.add(layers.LSTM(128, input_shape=(maxlen, len(chars))))
model.add(layers.Dense(len(chars), activation='softmax'))
```

```
In [5]: optimizer = keras.optimizers.RMSprop(lr=0.01)
model.compile(loss='categorical_crossentropy', optimizer=optimizer)
```

C:\Users\theoj\AppData\Roaming\Python\Python39\site-packages\keras\optimizers\optimizer_v2\rmsprop.py:140: UserWarning: The `lr` argument is deprecated, use `learning_rate` instead.
super().__init__(name, **kwargs)

```
In [6]: def sample(preds, temperature=1.0):
        preds = np.asarray(preds).astype('float64')
        preds = np.log(preds) / temperature
        exp_preds = np.exp(preds)
        preds = exp_preds / np.sum(exp_preds)
        probas = np.random.multinomial(1, preds, 1)
        return np.argmax(probas)
```

```

In [7]: import random
import sys

store_generated_text = ""
for epoch in range(1, 20):
    print('epoch', epoch)
    model.fit(x, y, batch_size=128, epochs=1)
    start_index = random.randint(0, len(text) - maxlen - 1)
    generated_text = text[start_index: start_index + maxlen]
    print('--- Generating with seed: "' + generated_text + '"')
    for temperature in [0.2, 0.5, 1.0, 1.2]:
        print('----- temperature:', temperature)
        sys.stdout.write(generated_text)
        for i in range(400):
            sampled = np.zeros((1, maxlen, len(chars)))
            for t, char in enumerate(generated_text):
                sampled[0, t, char_indices[char]] = 1.
            preds = model.predict(sampled, verbose=0)[0]
            next_index = sample(preds, temperature)
            next_char = chars[next_index]
            generated_text += next_char
            store_generated_text += next_char
            generated_text = generated_text[1:]
            sys.stdout.write(next_char)

```

```

of
the unordenousing our
past--without a personally me seems as the the. "ne tome
inrem
be centuries, and wsec can he wbind's beinged would bedol concerning. the would perhaper: for in be, as meta
physical to miffilir sey fellc man of omberion passy is brot, likewites from prtsfo'.ner
i.
wouds humbly than
year let us,
fellens dreepoch 9
1565/1565 [=====] - 12s 8ms/step - loss: 1.3889
--- Generating with seed: "thesis of a host of errors of the
intellect. to a world not "
----- temperature: 0.2
thesis of a host of errors of the

```

intellect. to a world not the conception and something that is the contemply to the subjection and something that is the same to the conception of the sense of the sense of the sense of the superficially to life of the present the sense and subjection of the sense of the signification of the sense of the same to the same to the subjection of the same to the present and subjection of the subjection of the subjection of the ----- temperature: 2.5

```
In [8]: f = open("results/generated.txt", "w")
        f.write(store_generated_text)
        f.close()
```