Modelling fonts with convolutional neural networks

Philip Bouman
10668667

***Abstract*—**
***Index Terms*—**

CONTENTS

INTRODUCTION

In most cases in natural languase there is no obvious relationship between the form and the meaning of a word. However in contrast to this arbitrariness there are instances of similarity between word form and meaning. A well studied phenomenon of such a relationship of iconicity is the *bouba-kiki* effect, in which shapes are consistently labeled with particular non-words. Iconic cross-sensory associations between sound properties and shape are generally accredited to be the source of this bias. Although culturally acquired ortographic influences may contribute to this bias aswell.

To gain a better insight to this entanglement of ortography and phonological features the goal of this research is to find if such relationships also exist between the shape and the sound of individual characters (letters). To avoid language-specific relations, a wide variety of writing systems (scripts) will be taken into consideration to discover general cross-cultural similarities between form and sound. To classify the sounds (phonemes) in a consistent way, a phonological encoding scheme will be used. This scheme contains all phonological features (whether a sound is nasal, labial, dorsal, etc.) of the phonemes as described in the International Phonetic Alphabet (IPA).

In the first part of the research a deep convolutional neural network will be trained to classify images of characters based on their related sounds. A deep convolutional neural network is a machine learning architecture consisting of multiple stacks of layers and has proved to be very succesfull for applications with two dimensional data. Furthermore machine learning allows a much wider range of writing systems to be analyzed than manually would be feasable.

The second part of the project will focus on getting a better understanding of the representations learned by the network. This will mainly be done by investigating and visualizing features learned in individual layers and smaller combinations of layers. This is done to provide an insight as to what specific properties of the characters are predictive phonological properties.

The results will be evaluated by measuring the level of accuracy on the test data.

LITERATURE REVIEW

The relation between word forms and their meanings in natural language is commonly referred to as sound symbolism. A well studied phenomenon of sound symbolism is the *bouba-kiki* effect, in which participants label abstract shapes with non-word names. Rounded shapes are systematically labelled bouba over kiki, this effect is often ascribed to cross-sensory mappings between acoustic properties of sound and shape. Cuskley et al. [1] suggest that this phenomenon is also heavily mediated by the symbolic, culturally acquired shapes of letters. This similarity between orthography and abstract shapes forms the basis of this research, in which the aim is to find if this relation can also be found between graphemes and phonemes. Graphemes represent the smallest individual characters of a writing system and phonemes relate to the sound of particular graphemes [2]. Phonemes represent a standardized collection of spoken language and are collected in the International Phonetic Alphabet (IPA) [3]. How these IPA symbols represent the exact pronunciation of a phoneme can be found in a phonological encoding scheme[1]. By using machine learning to classify graphemes to phonemes, a wide variety of writing systems[2] can be explored. These include Alphabets, Abjads (consonant alphabets) and Abugidas (symbols consisting of compositions of consonant and vowel). The desired machine learnng architecture will be a convolutional neural network (CNN), since CNNs are among the most suitable architectures for handwritten digit and character recognition [4] and not yet used for this specific problem. Neural networks learn intermediate representations of their inputs, which can be useful for subsequent classification tasks. The structure of CNNs,

[1] Appendix encoding scheme
[2] Appendix list of writing systems

mimicking receptive fields, is particularly useful for handling data that can be represented as two- or higher-dimensional input, such as images [5]. Deep CNNs also incorporate many layers and thus many intermediate representations, while keeping the number of free parameters small [6]. Recent work with CNNs on recognition of natural images and 3D objects resulted in higher accuracies than previously achieved by different methods. CNNs seem to work well for supervised and unsupervised learning tasks and can be succesfull using relatively small datasets [4].

RESEARCH QUESTION

Can a CNN succefully be used to classify graphemes of different writing systems to related general phonemes?

METHOD AND APPROACH

I. EVALUATION

The evaluation will be carried out by measuring the accuracy of the predicted phonemes for previously unseen graphemes using the test set. Depending on whether actual phoneme-grapheme relations exist, evalutation can be done on individual graphemes, simplified graphemes or grapheme clusters. Another part of the evaluation will consist of investigating the feature representations learned by the network. By visualising these representations they can be compared to other research concerning shape-sound relations.

II. APPROACH

A. Data

All of the initial data was collected from Omniglot[3], an online encyclopedia of writing systems and languages. This data, consisting of 350 different writing systems, was trimmed down to a set of 275 different scripts (see Appendix), due to either incomplete data or not being one of the categories (Abiguda, Abjad, Alphabet). For each language in the set, all the individual characters and their corresponding phoneme representations were extracted. These grapheme-phoneme pairs were then annotated by an expert to match IPA standards aswell as to reduce the number of different phonemes (from 2000 to 117 distinct phonemes).

The phonemes in the data represent the pronounciation of the individual characters, thus alterations of sounds when used in combinations with other characters are not considered. Characters with multiple phoneme options are split and added for each possibility (E.g.:' — ', which can be pronounced 'b' or 'p', will be added as two distinct entries: ' — ', [b] and ' — ', [p]).

Lastly the phoneme-grapheme pairs are converted to their final form to be used in the CNN. For every grapheme a greyscale image of 32 x 64 pixels is generated. Google NOTO fonts is used to achieve consistent layout between different characters.

Every phoneme/ IPA symbol is represented as a vector with fourteen features with positive, negative or absent values (+, -, 0). The following pronounciation features are considered: approximant, back, consonantal, continuant, delayed release, front, height, labial, low, nasal, rounded, sonorant, strident, tense.

From this data two different sets are created to be used in the model, both split three ways: training (80 %), validation (10 %) and test (10 %). The first data set is split language wise, keeping characters of the same language in the same set, the second set is split randomly on characters.

B. Architecture

The model was created with Keras using the Theano backend and the functional API class. The architecture of the network contains two convolutional layers and one fully-connected layer. The input of the model consist of 32 x 64 pixel greyscale images of characters and the output consists of fourteen layers for each pronounciation feature. Both convolutional layers are followed by a Max-pooling layer. The ReLU non-linearity is applied to the output of both convolutional layers.

III. EXPERIMENTS AND RESULTS

IV. DISCUSSION

V. CONCLUSION

VI. APPENDIX

A. Writing systems used

abaza
abenaki
abkhaz
acehnese
acheron
acholi
achuar-shiwiar
adamaua
adzera
afar
afrikaans
aghul
aguaruna
akan
akhvakh
aklan
akurio
alabama
albanian
alsatian
altay
alur
amahuaca
amarakaeri
andi
andoa
anuki
anutan
apache

[3]http://www.omniglot.com/charts/#xls

arabela
arabic-cypriot
arabic-msa
arabic-tunisian
arabic-turkic
arakanese
araki
aranese
arapaho
arawakan
archi
are
arikara
armenian
aromanian
arvanitic
arwi
ashaninka
asheninka
assamese
asturian
atlantean
avar
avestan
avokaya
awing
aymara
aynu
azeri
babine
badaga
bagvalal
balinese
balti
bambara
bandial
basque
beaver
bedik
beja
bench
bengali
bhojpuri
bislama
bisu
bora
borgu
bosnian
bouyei
brahmi
brahui
burmese
burushaski
busa
bushi
caddo
capeverdeancreole
caquinte
carian
catalan
caucasian
cayuga
chapalaa
chavacano
chechen
chickasaw
chilcotin
chipewyan
chuukese
cofan
comorian
comox
coptic
cuneiform
cyrillic-finnougaric
cyrillic-other
cyrillic-romance
cyrillic-russian
cyrillic-slavic
cyrillic-tungusic
cyrillic-turkic
dagaare
danish
degxinag
delaware
dutch
dzongkha
eskimo-aleut
estonian
ewondo
eyak
fijihindi
fula
futhorc
gitxsan
glagolitic
gothic
grantha
griko
guineabissaucreole
gujarati
hajong
hebrew
hieroglyphs
hindi
indonesian
interlingua
ipa
iranian
iroquoian
javanese
jeju

kabyle
kannada
karachay-balkar
karen
kashmiri
kayahli
kharosthi
khmer
khojki
khowar
korean
kove
kulitan
kumyk
kutchi
ladakhi
lao
latgalian
latin-aboriginal
latin-africa
latin-afroasiatic
latin-austroasiatic
latin-austronesian
latin-camerica
latin-celtic
latin-creoles
latin-english
latin-finnougaric
latin-formosan
latin-germanic
latin-hmongmien
latin-ial
latin-italic
latin-khoisan
latin-namerica
latin-nilosaharan
latin-samerica
latin-slavonic
latin-taikaidai
latin-tng
latin-turkic
latvian
lisu
lithuanian
lokoya
loma
lontara
lopit
lote
lycian
lydian
magahi
maithili
makonde
malay
malayalam
maltese
manchu
mandaic
mandarin
manipuri
marathi
marwari
maskelynes
mato
mayan
mende
mendekan
menominee
mongolic
monkhmer
mro
mutsun
nabataean
nadene
nepali
nheengatu
ocs
ogham
okinawan
oriya
oromo
pali
pawnee
phagspa
philippine
phonetic
pomoan
punjabi
quechuan
rarotongan
rejang
rohingya
romani
rovas
salishan
sankethi
sanskrit
shan
shina
siar
sikaiana
sikkimese
sinhala
sinitic
sio
somali
sundanese
sunuwar
sylheti
syriac
tami

tamil
teiwa
telugu
tengwar-arabic
tengwar-icelandic
tengwar-welsh
tengwar
thai
tibetan
tokipona
tongan
tsakonian
tshangla
tulu
tuvaluan
ubykh
uto-aztecan
wandamen
westernrote
wichita
wolof
yabem
zigula

B. Phonemes used

[nt]
[]
[]
[w]
[ts]
[h]
[Substitute]
[jæ]
[]
[d]
[œi]
[]
[k]
[oi]
[aj]
[v]
[iu]
[t]
[o]
[]
[ç]
[]
[ai]
[r]
[hv]
[y]
[p]
[i]
[o]
[e]
[]
[:]
[ea]
[g]
[w]
[õj]
[v]
[]
[mb]
[s]
[œu]
[p]
[]
[]
[]
[d]
[m]
[i]
[ja]
[ɪ]
[m]
[]
[i]
[p]
[ou]
[wo]
[SYLL]
[m]
[yi]
[w]
[ue]
[w]
[o]
[b]
[w]
[ũ]
[]
[j]
[b]
[l:]
[l]
[dz]
[]
[o]
[x]
[iu]
[ej]
[x]
[]
[c]
[ps]
[]
[]
[æ]
[nd]

[t]
[uai]
[n]
[j:]
[uo]
[ao]
[wi]
[j]
[syll]
[b]
[u]
[q]
[z]
[e]
[m]
[]
[nr]
[f]
[]
[d]
[]
[f]
[io]
[ui:]
[t]
[we]
[mv]
[]
[]
[e]
[]
[t]
[tl]
[q]
[p]
[r]
[s]
[]
[ua]
[t]
[eo]
[i]
[ae]
[i]
[]
[ts]
[s:]
[n]
[r]
[]
[u]
[]
[h]
[t]
[æi]
[k]
[kpr]
[]
[l]
[t]
[d]
[ç]
[l]
[]
[ui]
[y]
[œ]
[c]
[]
[]
[bv]
[øy]
[je]
[ld]
[]
[i]
[k]
[]
[]
[NA]
[œy]
[pf]
[]
[ũ:]
[k]
[s]
[]
[e]
[d]
[wæi]
[]
[r]
[]
[wa]
[u]
[e]
[]
[a]
[w]
[w]
[d]
[ie]
[]
[]
[d]
[w]
[ow]
[␣l]
[eu]
[w:]
[l]
[dz]

[Syll]
[j]
[ø]
[b]
[]
[nz]
[]
[j]
[m]
[:]
[ns]
[d]
[]
[u]
[j]
[u:]
[œ]
[d]
[ei]
[u]
[ts]
[]
[c]
[h]
[d]
[]
[]
[z]
[jo]
[v]
[ai]
[õ]
[oj]
[ju]
[t]
[]
[i]
[w]
[tr]
[]
[kx]
[a]
[au]
[a:]
[oi]
[kp]
[i]
[mp]
[]
[]
[uj]
[]
[y]
[]
[õ]
[aw]
[k]
[ia]
[i]
[]
[m]
[au]
[]
[i:]
[I]
[q]
[v]
[]
[st]
[d]
[]
[o:]
[f]
[s]

[consonantal, sonorant, continuant, delayedRelease, approximant, nasal, labial, rounded, strident, height, low, front, back, tense]

[0, 1, 1, 2, 1, 0, 0, 0, 2, 1, 1, 0, 1, 0]
[1, 1, 3, 2, 3, 3, 0, 0, 0, 0, 2, 2, 2, 2]
[1, 1, 1, 2, 1, 0, 0, 0, 0, 0, 2, 2, 2, 2]
[3, 0, 1, 1, 0, 0, 0, 3, 2, 0, 2, 2, 2, 2]
[1, 0, 0, 0, 0, 0, 0, 0, 2, 1, 0, 0, 0, 1]
[0, 1, 1, 2, 1, 0, 0, 3, 2, 1, 0, 0, 3, 3]
[1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 2, 2, 2, 2]
[0, 1, 1, 2, 1, 0, 0, 1, 2, 1, 1, 0, 1, 0]
[0, 1, 1, 2, 1, 0, 0, 1, 2, 1, 3, 0, 3, 3]
[1, 3, 3, 0, 3, 3, 0, 0, 0, 0, 2, 2, 2, 2]
[0, 1, 1, 2, 1, 0, 0, 1, 2, 1, 3, 0, 1, 0]
[1, 0, 0, 0, 0, 0, 0, 1, 2, 1, 0, 0, 0, 1]
[1, 1, 0, 2, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0]
[1, 1, 0, 2, 0, 0, 0, 1, 2, 1, 1, 0, 0, 0]
[1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 2, 2, 2, 2]
[1, 3, 3, 0, 3, 0, 3, 0, 0, 0, 2, 2, 2, 2]
[1, 0, 1, 1, 0, 0, 0, 1, 2, 1, 1, 0, 0, 0]
[0, 1, 1, 2, 1, 0, 0, 1, 2, 1, 3, 0, 0, 1]
[0, 1, 1, 2, 1, 0, 0, 0, 2, 1, 0, 1, 0, 1]
[0, 1, 1, 2, 1, 0, 0, 3, 2, 1, 1, 0, 3, 3]
[0, 1, 1, 2, 1, 0, 0, 1, 2, 1, 0, 0, 0, 1]
[1, 3, 3, 1, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2]
[1, 3, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2]
[1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 2, 2, 2, 2]
[1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0]
[0, 1, 1, 2, 1, 0, 0, 1, 2, 1, 0, 0, 1, 0]
[1, 0, 0, 1, 0, 0, 0, 1, 2, 0, 2, 2, 2, 2]
[1, 0, 1, 1, 0, 0, 0, 0, 2, 1, 0, 1, 0, 1]
[0, 0, 1, 1, 0, 0, 0, 1, 2, 0, 2, 2, 2, 2]
[0, 1, 1, 2, 1, 0, 0, 3, 2, 1, 3, 3, 3, 0]
[1, 3, 0, 0, 0, 0, 0, 1, 2, 0, 2, 2, 2, 2]
[1, 1, 1, 2, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1]

```
[1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 2, 2, 2, 2]
[1, 0, 1, 1, 0, 0, 0, 1, 2, 1, 0, 1, 0, 1]
[0, 0, 1, 1, 0, 0, 0, 0, 2, 0, 2, 2, 2, 2]
[0, 1, 1, 2, 1, 0, 0, 3, 2, 1, 3, 0, 1, 0]
[0, 1, 1, 2, 1, 0, 0, 3, 2, 1, 3, 0, 3, 0]
[0, 1, 1, 2, 1, 0, 0, 3, 2, 1, 3, 3, 0, 3]
[1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0]
[1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0]
[1, 0, 3, 3, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2]
[0, 1, 1, 2, 1, 0, 0, 1, 2, 1, 0, 0, 0, 0]
[0, 1, 1, 2, 1, 0, 0, 0, 2, 1, 1, 0, 0, 0]
[0, 1, 1, 2, 1, 0, 0, 3, 2, 1, 3, 0, 3, 3]
[1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 2, 2, 2, 2]
[1, 3, 0, 0, 0, 0, 0, 0, 2, 1, 1, 0, 0, 0]
[0, 1, 1, 2, 1, 0, 0, 0, 2, 1, 3, 3, 1, 0]
[1, 0, 1, 1, 0, 0, 0, 1, 2, 1, 1, 0, 1, 1]
[0, 1, 1, 2, 1, 0, 0, 0, 2, 1, 0, 1, 0, 0]
[0, 1, 1, 2, 1, 0, 0, 0, 2, 1, 0, 0, 1, 0]
[1, 0, 0, 0, 0, 0, 0, 1, 2, 0, 2, 2, 2, 2]
[1, 0, 0, 0, 0, 0, 0, 1, 2, 1, 1, 0, 0, 0]
[1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2]
[1, 1, 1, 2, 1, 0, 1, 0, 0, 0, 2, 2, 2, 2]
[0, 1, 1, 2, 1, 0, 0, 0, 2, 1, 0, 0, 0, 0]
[1, 1, 0, 2, 0, 0, 0, 1, 2, 0, 2, 2, 2, 2]
[1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 2, 2, 2, 2]
[1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1]
[1, 1, 0, 2, 0, 0, 0, 0, 2, 1, 1, 0, 0, 0]
[1, 0, 0, 1, 0, 0, 0, 0, 2, 1, 1, 0, 0, 0]
[0, 1, 1, 2, 1, 0, 0, 0, 2, 1, 3, 0, 3, 0]
[1, 0, 0, 0, 0, 0, 0, 1, 2, 1, 1, 0, 1, 1]
[0, 1, 1, 2, 1, 0, 0, 0, 2, 1, 3, 3, 3, 0]
[1, 3, 3, 1, 0, 0, 0, 1, 2, 0, 2, 2, 2, 2]
[1, 0, 0, 0, 0, 0, 0, 0, 2, 1, 1, 0, 1, 1]
[0, 1, 1, 2, 1, 0, 0, 0, 2, 1, 0, 3, 3, 0]
[0, 1, 1, 2, 1, 0, 0, 0, 2, 1, 3, 0, 1, 0]
[0, 1, 1, 2, 1, 0, 0, 0, 2, 1, 0, 1, 1, 0]
[0, 1, 1, 2, 1, 0, 0, 0, 2, 1, 0, 0, 3, 0]
[1, 0, 0, 0, 0, 0, 0, 0, 2, 1, 1, 0, 0, 0]
[0, 1, 1, 2, 1, 0, 0, 1, 2, 0, 2, 2, 2, 2]
[1, 0, 3, 3, 0, 0, 0, 3, 0, 0, 2, 2, 2, 2]
[0, 1, 1, 2, 1, 0, 0, 3, 2, 1, 3, 0, 0, 3]
[1, 0, 1, 1, 0, 0, 0, 0, 2, 1, 1, 0, 0, 0]
[1, 3, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0]
[1, 0, 0, 0, 0, 0, 0, 0, 2, 0, 2, 2, 2, 2]
[0, 0, 1, 1, 0, 0, 0, 0, 2, 1, 1, 0, 1, 1]
[0, 1, 1, 2, 1, 0, 0, 0, 2, 1, 1, 0, 0, 1]
[1, 1, 0, 2, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2]
[1, 0, 1, 1, 0, 0, 0, 0, 2, 1, 0, 0, 0, 1]
[1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2]
[3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3]
[0, 1, 1, 2, 1, 0, 0, 1, 2, 1, 1, 0, 0, 1]
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2]
[1, 0, 1, 1, 0, 0, 0, 1, 2, 0, 2, 2, 2, 2]
[0, 1, 1, 2, 1, 0, 0, 3, 2, 1, 0, 3, 0, 3]
[1, 3, 0, 0, 0, 0, 0, 0, 2, 1, 0, 0, 0, 1]
[1, 1, 1, 2, 1, 1, 0, 0, 0, 0, 2, 2, 2, 2]
[1, 1, 0, 2, 0, 0, 0, 3, 2, 3, 1, 0, 0, 0]
[0, 1, 1, 2, 1, 0, 0, 3, 2, 1, 3, 3, 3, 3]
[1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0]
[1, 1, 1, 2, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1]
```

REFERENCES

[1] Christine Cuskley, Julia Simner, and Simon Kirby. Phonological and orthographic influences in the bouba–kiki effect. *Psychological Research*, 81(1):119–130, 2017.

[2] Mark S Seidenberg. Beyond orthographic depth in reading: Equitable division of labor. *Advances in psychology*, 94:85–118, 1992.

[3] International Phonetic Association. *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press, 1999.

[4] Dan Claudiu Ciresan, Ueli Meier, Luca Maria Gambardella, and Jurgen Schmidhuber. Convolutional neural network committees for handwritten character classification. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 1135–1139. IEEE, 2011.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[6] Théodore Bluche, Hermann Ney, and Christopher Kermorvant. Feature extraction with convolutional neural networks for handwritten word recognition. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 285–289. IEEE, 2013.