ML4AAD - Final Project

Winter Semeseter 18/19

Neeratyoy Mallik

4774378

February 24, 2019

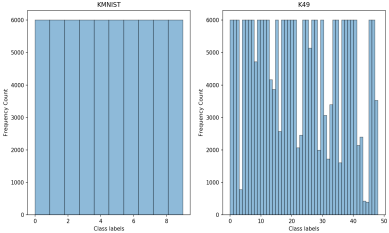
Motivation (The Why)

AAAAAHH!!
WWW.PHDCOMICS.COM
sip...
aahh..

Motivation (The Why)

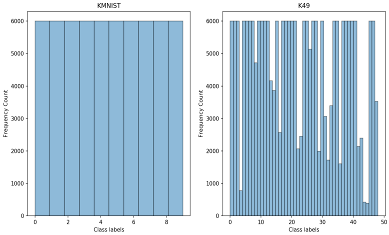
Motivation (The Why)

2 (similar) datasets with differing size, # of classes, class balance

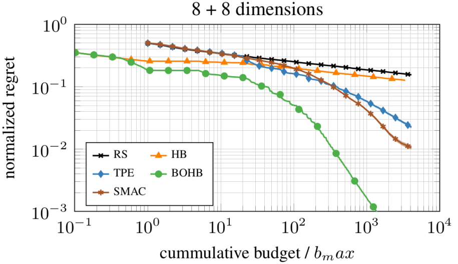


Motivation (The Why)

2 (similar) datasets with differing size, # of classes, class balance



BOHB outperforms SMAC in high dimensional, mixed data



Design Decisions (and BOHB)

CNN Structure

$INPUT \longrightarrow [CONV \rightarrow BATCHNORM? \rightarrow ACTIVATION \rightarrow DROPOUT? \rightarrow MAXPOOL?]*M \longrightarrow [FC \rightarrow BATCHNORM? \rightarrow ACTIVATION \rightarrow DROPOUT?]*K \longrightarrow OUTPUT$

$M \in \{1,2,3\}; K \in \{0,1,2\}; ? \rightarrow \top or \bot$

Design Decisions (and BOHB)

CNN Structure

$INPUT \longrightarrow [CONV \rightarrow BATCHNORM? \rightarrow ACTIVATION \rightarrow DROPOUT? \rightarrow MAXPOOL?]*M \longrightarrow [FC \rightarrow BATCHNORM? \rightarrow ACTIVATION \rightarrow DROPOUT?]*K \longrightarrow OUTPUT$
$M \in \{1, 2, 3\}$; $K \in \{0, 1, 2\}$; $? \rightarrow \top or \bot$

- CONVolution layers
  - Kernel size
  - Padding
  - Stride
- ACTIVATION (relu/sigmoid/tanh)
- BATCHNORM, DROPOUT
  - True or False
- MAXPOOL (if True)
  - Kernel size (=stride)

Design Decisions (and BOHB)

CNN Structure

$INPUT \longrightarrow [CONV \rightarrow BATCHNORM? \rightarrow ACTIVATION \rightarrow DROPOUT? \rightarrow MAXPOOL?]*M \longrightarrow [FC \rightarrow BATCHNORM? \rightarrow ACTIVATION \rightarrow DROPOUT?]*K \longrightarrow OUTPUT$
$M \in \{1, 2, 3\}$; $K \in \{0, 1, 2\}$; $? \rightarrow \top or \bot$

- CONVolution layers
  - Kernel size
  - Padding
  - Stride
- ACTIVATION (relu/sigmoid/tanh)
- BATCHNORM, DROPOUT
  - True or False
- MAXPOOL (if True)
  - Kernel size (=stride)
- Learning rate
- Optimizer
- Batch size

Design Decisions (and BOHB)

CNN Structure

$INPUT \rightarrow [CONV \rightarrow BATCHNORM? \rightarrow ACTIVATION \rightarrow DROPOUT? \rightarrow MAXPOOL?]*M \rightarrow [FC \rightarrow BATCHNORM? \rightarrow ACTIVATION \rightarrow DROPOUT?]*K \rightarrow OUTPUT$
$M \in \{1,2,3\}$; $K \in \{0,1,2\}$; $? \rightarrow \top or \bot$

- CONVolution layers
  - Kernel size
  - Padding
  - Stride
- ACTIVATION (relu/sigmoid/tanh)
- BATCHNORM, DROPOUT
  - True or False
- MAXPOOL (if True)
  - Kernel size (=stride)
- Learning rate
- Optimizer
- Batch size

- >30 hyperparameters
- (Budget = Epochs) → expensive runs

Design Decisions (and BOHB)

CNN Structure

$INPUT \longrightarrow [CONV \rightarrow BATCHNORM? \rightarrow ACTIVATION \rightarrow DROPOUT? \rightarrow MAXPOOL?]*M \longrightarrow [FC \rightarrow BATCHNORM? \rightarrow ACTIVATION \rightarrow DROPOUT?]*K \longrightarrow OUTPUT$
$M \in \{1, 2, 3\}$; $K \in \{0, 1, 2\}$; $? \rightarrow \top or \bot$

- CONVolution layers
  - Kernel size
  - Padding
  - Stride
- ACTIVATION (relu/sigmoid/tanh)
- BATCHNORM, DROPOUT
  - True or False
- MAXPOOL (if True)
  - Kernel size (=stride)
- Learning rate
- Optimizer
- Batch size

- $>$30 hyperparameters
- (Budget = Epochs) → expensive runs

BOHB params:

eta	min_budget	max_budget
2	1	16
4	1	16
3	1	9
2	1	10

BOHB on KMNIST (and K49)

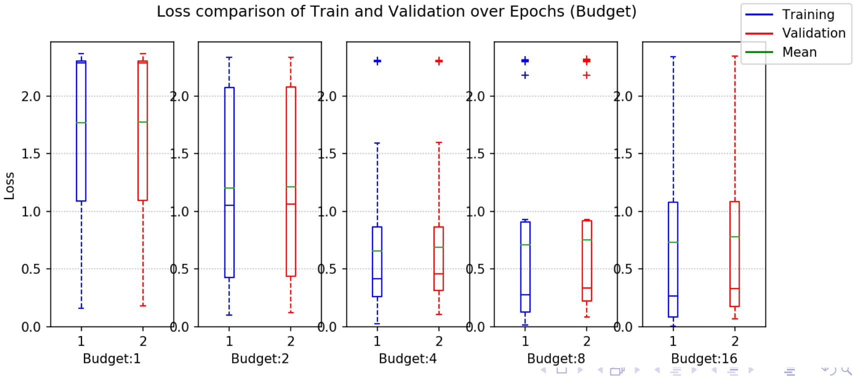
Dataset	eta_min_max_iter (BOHB)	Validation Accuracy	Train Accuracy	Test Accuracy	BOHB Runtime
KMNIST	2_1_16_10	98.08%	99.81%	95.06%	<3 hrs
KMNIST	3_1_9_20	97.17%	98.23%	93.26%	<3 hrs
KMNIST	4_1_16_20	98.26%	99.98%	96.41%	<4 hrs
KMNIST	2_1_10_14	97.77%	99.86%	94.08%	<4 hrs
KMNIST	3_2_20_20	98.31%	99.62%	95.41%	<6 hrs

BOHB on KMNIST (and K49)

Dataset	eta_min_max_iter (BOHB)	Validation Accuracy	Train Accuracy	Test Accuracy	BOHB Runtime
KMNIST	2_1_16_10	98.08%	99.81%	95.06%	<3 hrs
KMNIST	3_1_9_20	97.17%	98.23%	93.26%	<3 hrs
KMNIST	4_1_16_20	98.26%	99.98%	96.41%	<4 hrs
KMNIST	2_1_10_14	97.77%	99.86%	94.08%	<4 hrs
KMNIST	3_2_20_20	98.31%	99.62%	95.41%	<6 hrs
K49	3_1_9_10	89.06%	99.93%	88.12%	<5 hrs
K49	2_1_10_10	91.19%	99.25%	88.25%	<12 hrs

BOHB on KMNIST (and K49)

Dataset	eta_min_max_iter (BOHB)	Validation Accuracy	Train Accuracy	Test Accuracy	BOHB Runtime
KMNIST	2_1_16_10	98.08%	99.81%	95.06%	<3 hrs
KMNIST	3_1_9_20	97.17%	98.23%	93.26%	<3 hrs
KMNIST	4_1_16_20	98.26%	99.98%	96.41%	<4 hrs
KMNIST	2_1_10_14	97.77%	99.86%	94.08%	<4 hrs
KMNIST	3_2_20_20	98.31%	99.62%	95.41%	<6 hrs
K49	3_1_9_10	89.06%	99.93%	88.12%	<5 hrs
K49	2_1_10_10	91.19%	99.25%	88.25%	<12 hrs



Extracting juice (Can K49 leverage KMNIST?)

Model: KMNIST → K49

KMNIST Test	K49 Train	K49 Test	Run-time
95.06%	97.15%	90.49%	<2hrs
95.41%	95.91%	90.63%	<1hr
96.41%	99.67%	93.32%	<4hrs

Extracting juice (Can K49 leverage KMNIST?)

Model: KMNIST → K49

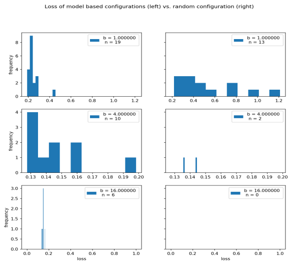
KMNIST Test	K49 Train	K49 Test	Run-time
95.06%	97.15%	90.49%	<2hrs
95.41%	95.91%	90.63%	<1hr
96.41%	99.67%	93.32%	<4hrs

Configuration: KMNIST $\xrightarrow{\text{BOHB}}$ K49

Hyperparameters:

- # of channels
- # of FC layers and neurons
- batch size

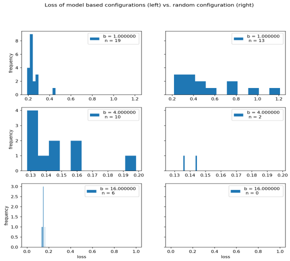
KMNIST Test	K49 Train	K49 Test	Run-time
95.41%	98.87%	90.20%	<4hrs
96.41%	99.93%	94.28%	<20hrs



Extracting juice (Can K49 leverage KMNIST?)

Model: KMNIST → K49

KMNIST Test	K49 Train	K49 Test	Run-time
95.06%	97.15%	90.49%	<2hrs
95.41%	95.91%	90.63%	<1hr
96.41%	99.67%	93.32%	<4hrs

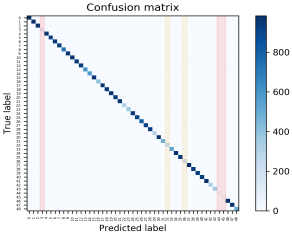


Configuration: KMNIST $\xrightarrow{\text{BOHB}}$ K49

Hyperparameters:

- # of channels
- # of FC layers and neurons
- batch size

KMNIST Test	K49 Train	K49 Test	Run-time
95.41%	98.87%	90.20%	<4hrs
96.41%	99.93%	94.28%	<20hrs

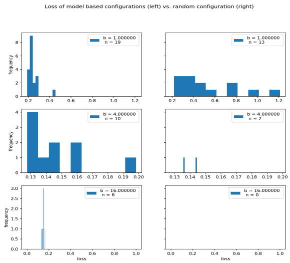


Issues: (i) Slow experiments; (ii) Under-represented classes;

Extracting juice (Can K49 leverage KMNIST?)

Model: KMNIST → K49

KMNIST Test	K49 Train	K49 Test	Run-time
95.06%	97.15%	90.49%	<2hrs
95.41%	95.91%	90.63%	<1hr
96.41%	99.67%	93.32%	<4hrs

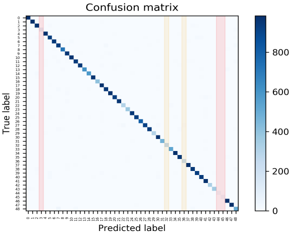

Loss of model based configurations (left) vs. random configuration (right)

Configuration: KMNIST $\xrightarrow{\text{BOHB}}$ K49

Hyperparameters:

- # of channels
- # of FC layers and neurons
- batch size

KMNIST Test	K49 Train	K49 Test	Run-time
95.41%	98.87%	90.20%	<4hrs
96.41%	99.93%	94.28%	<20hrs



Issues: (i) Slow experiments; (ii) Under-represented classes; [(iii) What if only K49?]

In Conclusion...

Possible improvements:

- Fidelity (data subsets)
    - K49: 90.20% → 93.85%

In Conclusion...

Possible improvements:

- Fidelity (data subsets)
  - K49: 90.20% → 93.85%
- Data Augmentation
  - K49: 94.28% → 93.37% (with fidelity)

In Conclusion...

Possible improvements:

- Fidelity (data subsets)
  - K49: 90.20% → 93.85%
- Data Augmentation
  - K49: 94.28% → 93.37% (with fidelity)
- Better/more parameters for transfer configuration
  - Learning rate schedule
  - Optimizer

In Conclusion...

Possible improvements:

- Fidelity (data subsets)
  - K49: 90.20% → 93.85%
- Data Augmentation
  - K49: 94.28% → 93.37% (with fidelity)
- Better/more parameters for transfer configuration
  - Learning rate schedule
  - Optimizer
- EPMs to try reduce parameter blow ups and runtimes

In Conclusion...

Possible improvements:

- Fidelity (data subsets)
  - K49: 90.20% → 93.85%
- Data Augmentation
  - K49: 94.28% → 93.37% (with fidelity)
- Better/more parameters for transfer configuration
  - Learning rate schedule
  - Optimizer
- EPMs to try reduce parameter blow ups and runtimes
- NAS cell search

In Conclusion...

Possible improvements:

- Fidelity (data subsets)
  - K49: 90.20% → 93.85%
- Data Augmentation
  - K49: 94.28% → 93.37% (with fidelity)
- Better/more parameters for transfer configuration
  - Learning rate schedule
  - Optimizer
- EPMs to try reduce parameter blow ups and runtimes
- NAS cell search
- Padding to go deeper (exploit sparsity)

In Conclusion...

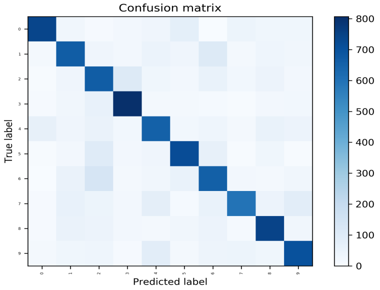
Possible improvements:

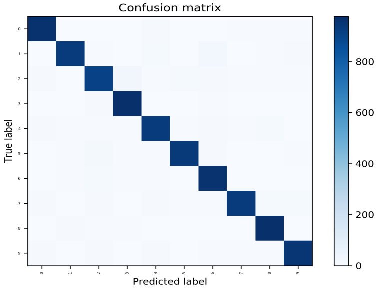
- Fidelity (data subsets)
  - K49: 90.20% → 93.85%
- Data Augmentation
  - K49: 94.28% → 93.37% (with fidelity)
- Better/more parameters for transfer configuration
  - Learning rate schedule
  - Optimizer
- EPMs to try reduce parameter blow ups and runtimes
- NAS cell search
- Padding to go deeper (exploit sparsity)

Final Numbers:

Dataset	Default Test	Best Test
KMNIST	69.56%	97.89%
K49	54.43%	94.28%

KMNIST: *from transferring best K49 model*
K49: *from transferring incumbent config on KMNIST and then BOHB on K49*


Confusion matrix
True label
Predicted label


Confusion matrix
True label
Predicted label