Capsule Neural Networks for Text Classification

Akmal Khikmatullaev, 2745116

Outline

- Introduction
- CapsNet with DR(Dynamic Routing)
- CapsNet with EMR(EM Routing)
- Approach
- Evaluation
- Conclusion

Text Classification(TC)

Introduction

$\mathcal{D}$ – a domain of documents
$\mathcal{C} = \{c_1, c_2, \ldots c_{|\mathcal{C}|}\}$ – a set of pre-defined *categories*

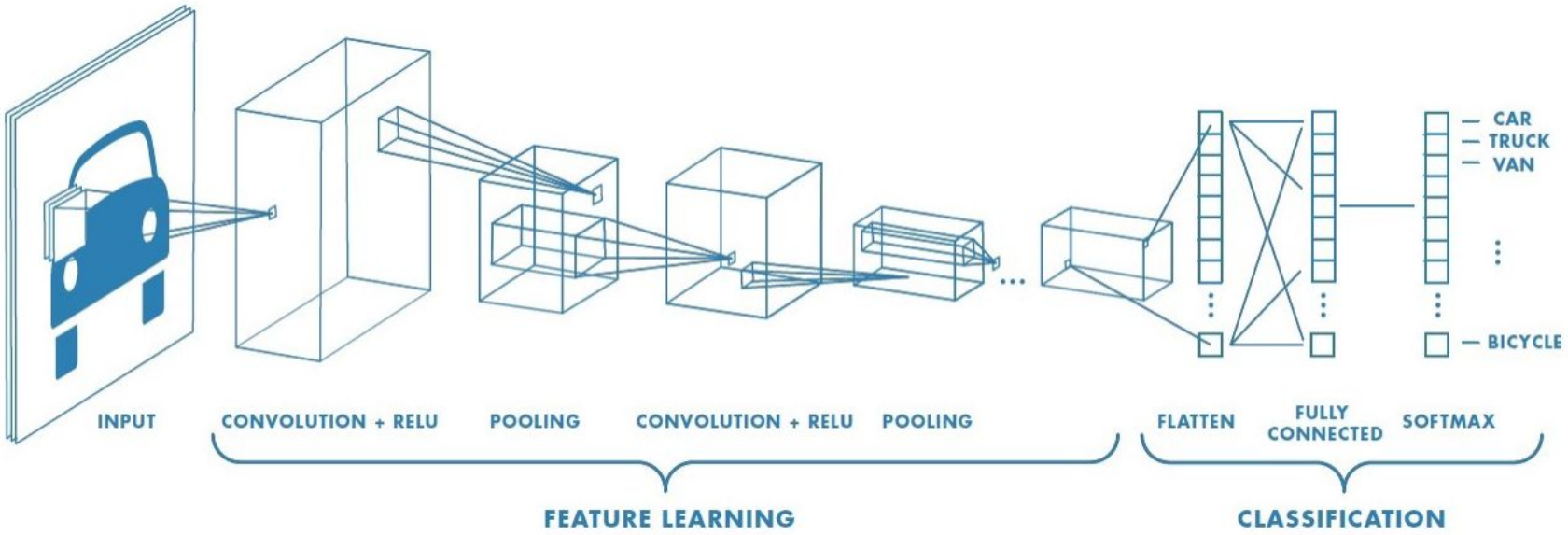
$\forall d \in \mathcal{D}$ assign a Boolean value $b_i \in \{0, 1\}, \forall \langle d, c_i \rangle \in \mathcal{D} \times \mathcal{C} \Leftrightarrow$

$$\begin{aligned} \langle d, c_1 \rangle &= b_1 \\ \langle d, c_2 \rangle &= b_2 \\ &\ldots \\ \langle d, c_{|\mathcal{C}|} \rangle &= b_{|\mathcal{C}|} \end{aligned}$$

More formally, this is the task of a ***target function approximation***:

$$\Phi : \mathcal{D} \times \mathcal{C} \rightarrow \{0, 1\}$$

Convolutional Neural Network(CNN) Introduction



CNN for TC

Introduction

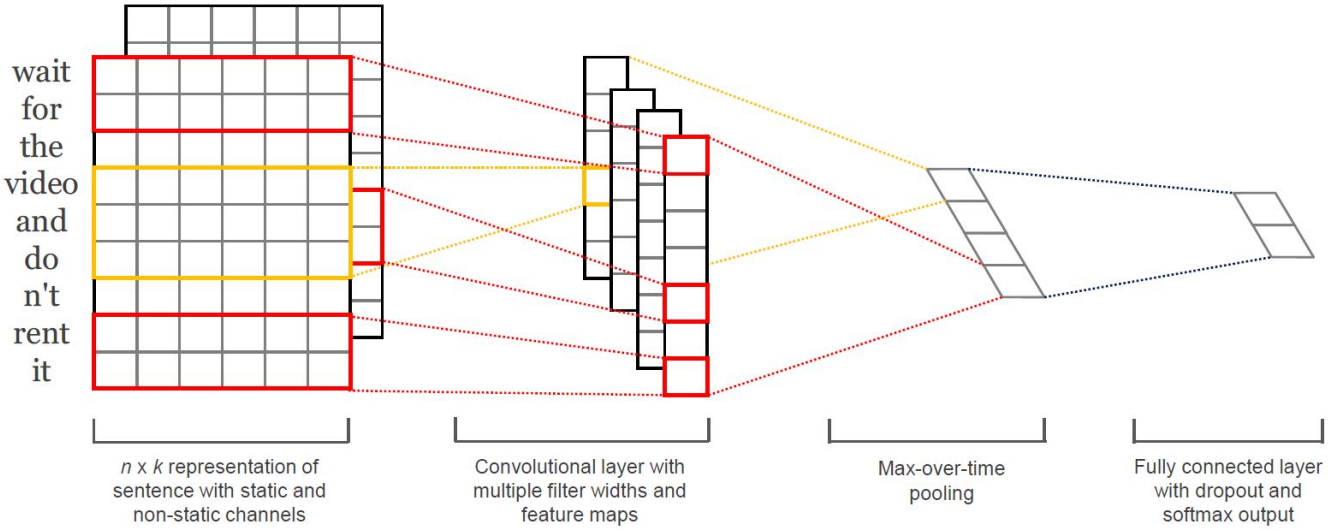
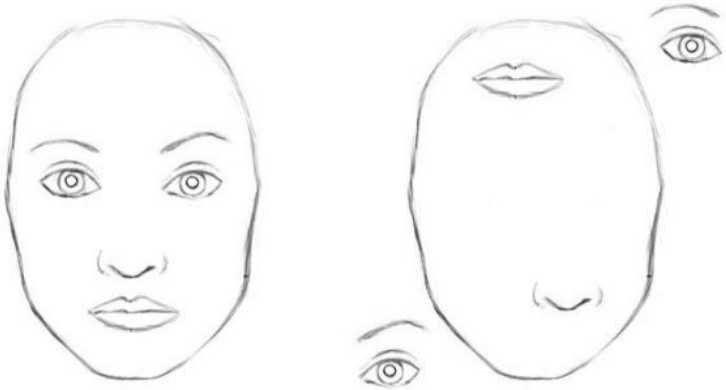
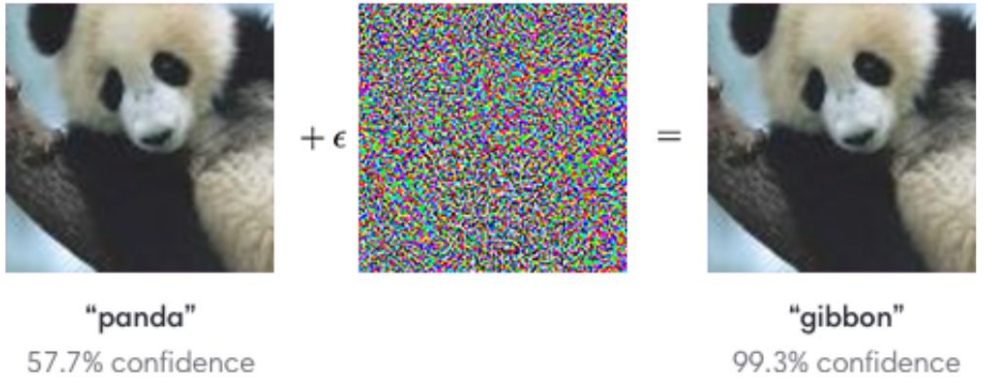


Figure 1: Model architecture with two channels for an example sentence.

CNN Drawbacks



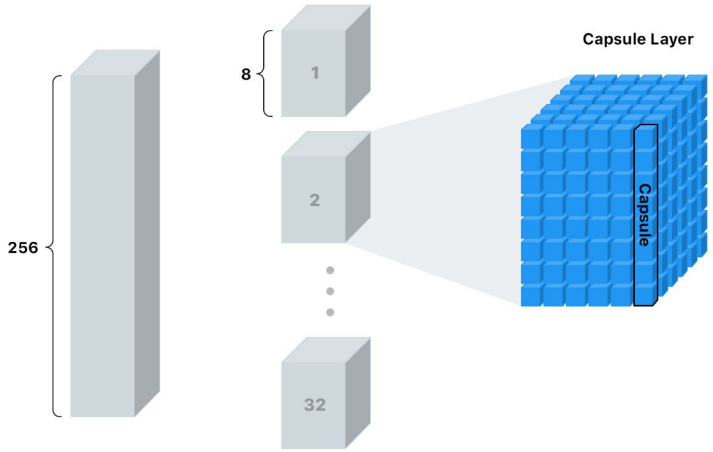
Introduction



Capsule (1)

CapsNet with DR

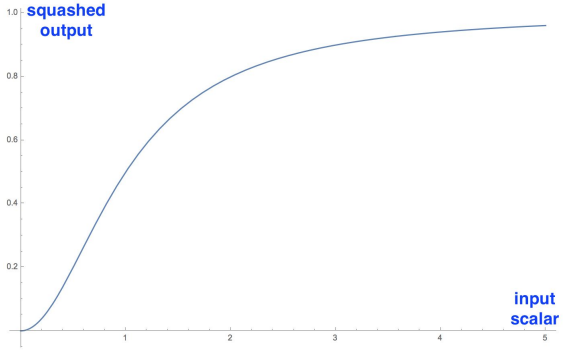
- ***Capsule*** - a group of neurons, the activity vector represents an object part or an object itself
- Capsule ***encapsulates*** all important information about the state of the feature it is detecting in vector form
- The ***length*** of this vector is a probability of a feature detection



Capsule (2)

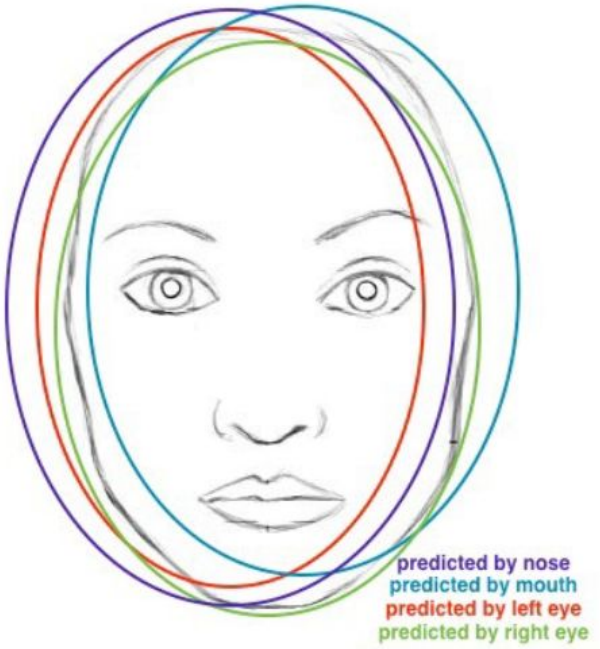
CapsNet with DR

Capsule vs. Traditional Neuron			
Input from low-level capsule/neuron		vector($\mathbf{u}_i$)	scalar(x_i)
Operation	Affine Transform	$\widehat{\mathbf{u}}_{j\|i} = \mathbf{W}_{ij}\mathbf{u}_i$	–
	Weighting	$\mathbf{s}_j = \sum_i c_{ij}\widehat{\mathbf{u}}_{j\|i}$	$a_j = \sum_i w_i x_i + b$
	Sum		
	Nonlinear Activation	$\mathbf{v}_j = \frac{\|\mathbf{s}_j\|^2}{1+\|\mathbf{s}_j\|^2}\frac{\mathbf{s}_j}{\|\mathbf{s}_j\|}$	$h_j = f(a_j)$
Output		vector($\mathbf{v}_j$)	scalar(h_j)



Capsule (3)

CapsNet with DR



Predictions for face location of nose, mouth and eyes capsules closely match: there must be a face there.

Dynamic routing (1)

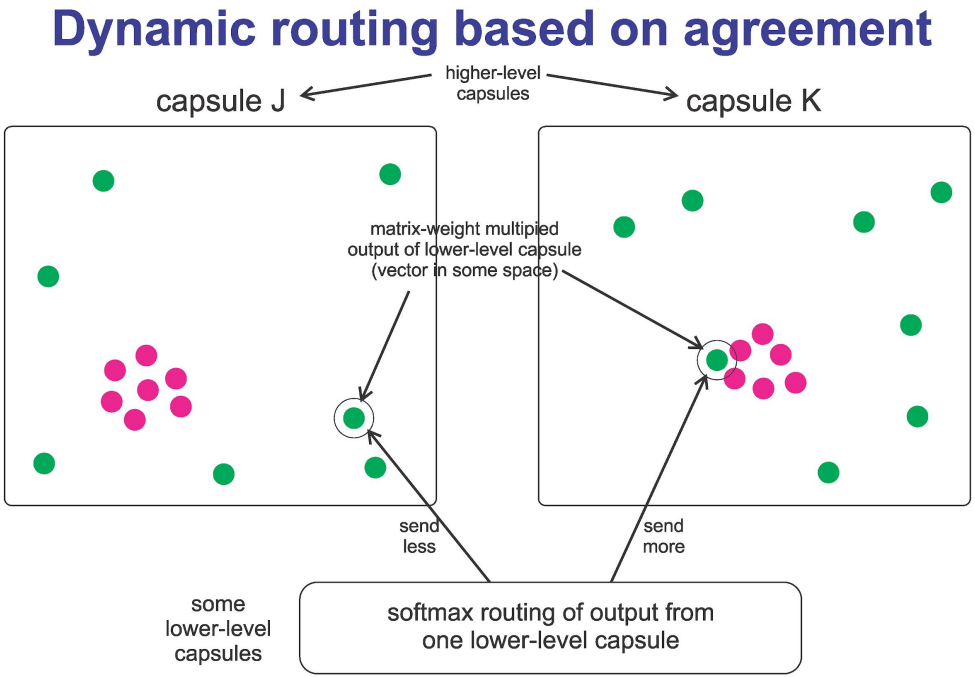
CapsNet with DR

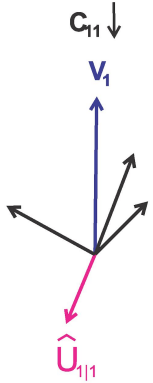
Procedure 1 Routing algorithm.

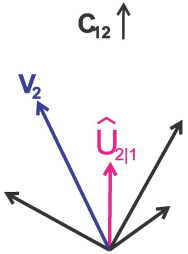
1: **procedure** ROUTING($\hat{\boldsymbol{u}}_{j|i}, r, l$)
2: for all capsule i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow 0$.
3: **for** r iterations **do**
4: for all capsule i in layer l: $\mathbf{c}_i \leftarrow \texttt{softmax}(\mathbf{b}_i)$ ▷ `softmax` computes Eq. 3
5: for all capsule j in layer $(l+1)$: $\mathbf{s}_j \leftarrow \sum_i c_{ij} \hat{\mathbf{u}}_{j|i}$
6: for all capsule j in layer $(l+1)$: $\mathbf{v}_j \leftarrow \texttt{squash}(\mathbf{s}_j)$ ▷ `squash` computes Eq. 1
7: for all capsule i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow b_{ij} + \hat{\mathbf{u}}_{j|i}.\mathbf{v}_j$
return $\mathbf{v}_j$

Dynamic routing (2)

CapsNet with DR

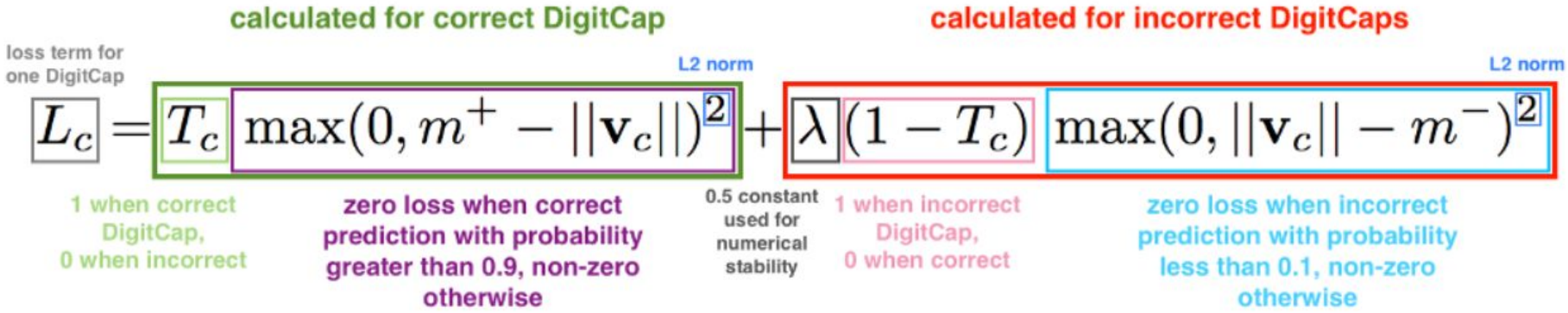






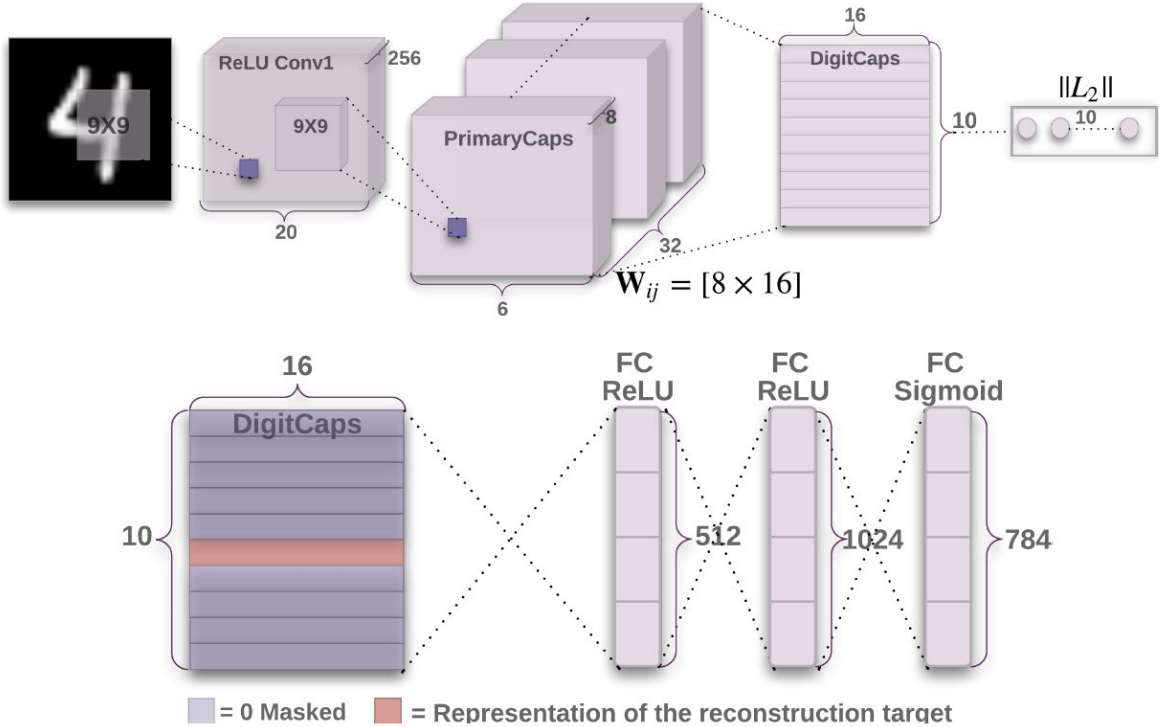
Loss Function

CapsNet with DR



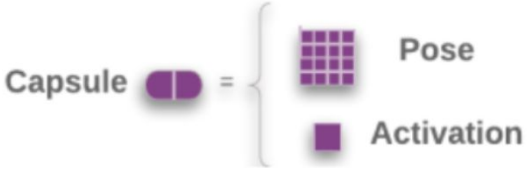
Architecture

CapsNet with DR



Matrix Capsule

Matrix capsule - the new type of capsules represents an entity and 4x4 matrix which is able to find between to an entity and the viewer(the pose) the relationship.



CapsNet with EMR

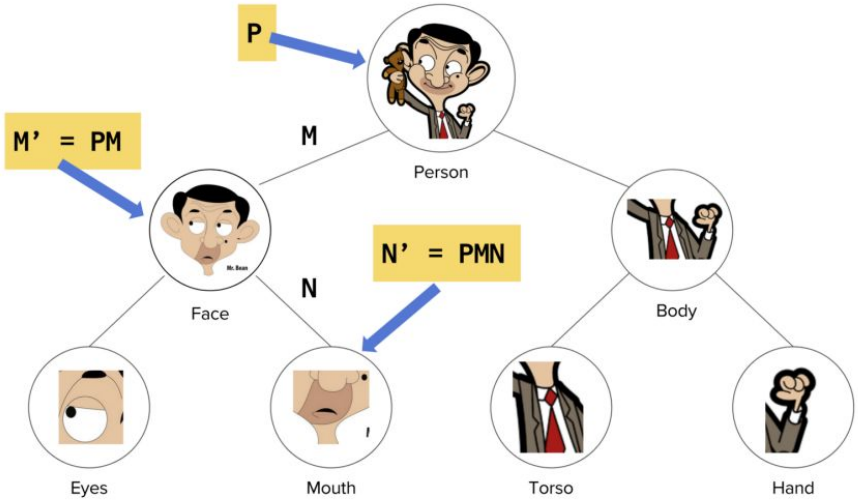
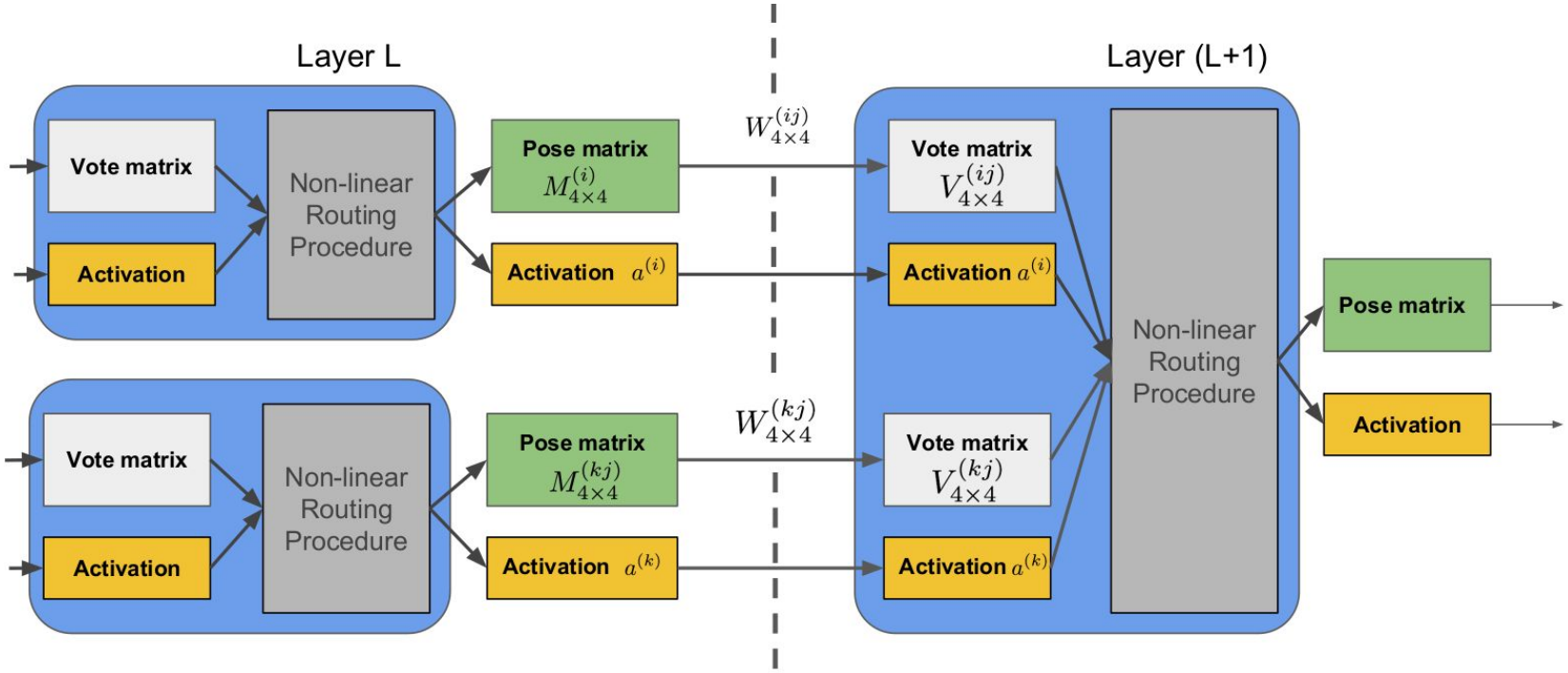


Figure 1: Pose matrices representing the hierarchical relationship.
P: pose matrix of the person.
M: spatial relationship between face and person
N: represents the relationship between mouth and face.
M' and N': pose matrices for the face and the mouth

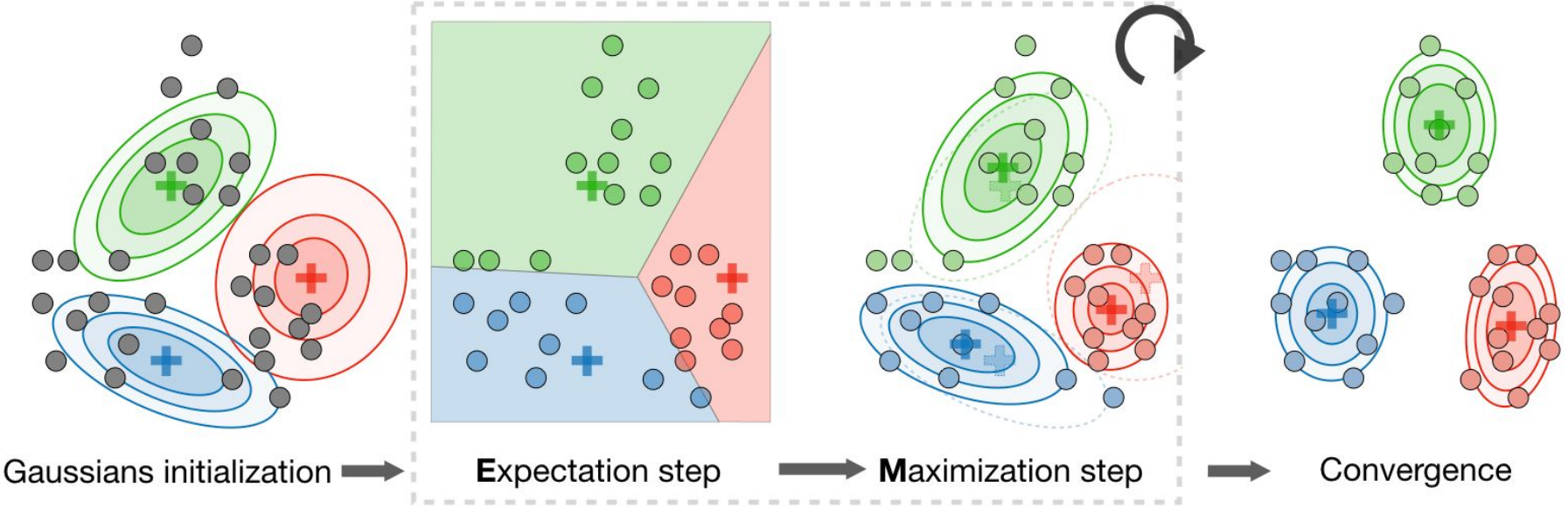
EM Routing (1)

CapsNet with EMR



GMM: EM-Algorithm

CapsNet with EMR



EM Routing (2)

CapsNet with EMR

Procedure 1 Routing algorithm

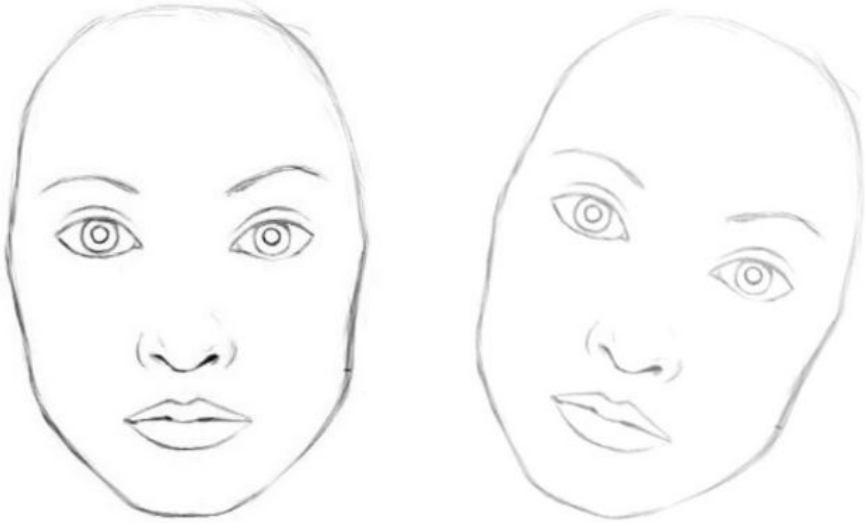
procedure EM ROUTING($\boldsymbol{a}, V$)
  $\forall i \in \Omega_L, j \in \Omega_{L+1}$: $R_{ij} \leftarrow 1/|\Omega_{L+1}|$
  for t iterations **do**
    $\forall j \in \Omega_{L+1}$: M-STEP($\boldsymbol{a}, R, V, j$)
    $\forall i \in \Omega_L$: E-STEP($\mu, \sigma, \boldsymbol{a}, V, i$)
  return $\boldsymbol{a}, M$

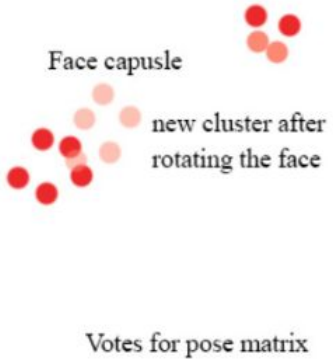
procedure M-STEP($\boldsymbol{a}, R, V, j$) ▷ for one higher-level capsule, j
  $\forall i \in \Omega_L$: $R_{ij} \leftarrow R_{ij} * \boldsymbol{a}_i$
  $\forall h$: $\mu_j^h \leftarrow \frac{\sum_i R_{ij} V_{ij}^h}{\sum_i R_{ij}}$
  $\forall h$: $(\sigma_j^h)^2 \leftarrow \frac{\sum_i R_{ij} (V_{ij}^h - \mu_j^h)^2}{\sum_i R_{ij}}$
  $cost^h \leftarrow \left(\beta_u + log(\sigma_j^h)\right) \sum_i R_{ij}$
  $a_j \leftarrow logistic(\lambda(\beta_a - \sum_h cost^h))$

procedure E-STEP($\mu, \sigma, \boldsymbol{a}, V, i$) ▷ for one lower-level capsule, i
  $\forall j \in \Omega_{L+1}$: $\boldsymbol{p}_j \leftarrow \frac{1}{\sqrt{\prod_h^H 2\pi(\sigma_j^h)^2}} \exp\left(-\sum_h^H \frac{(V_{ij}^h - \boldsymbol{\mu}_j^h)^2}{2(\sigma_j^h)^2}\right)$
  $\forall j \in \Omega_{L+1}$: $\boldsymbol{R}_{ij} \leftarrow \frac{\boldsymbol{a}_j p_j}{\sum_{k \in \Omega_{L+1}} \boldsymbol{a}_k p_k}$

EM Routing (3)

CapsNet with EMR





Loss Function

CapsNet with EMR

target class's activation

L2 norm

total loss

$$L_i = (max(0, m - (a_t - a_i))^2, \quad L = \sum_{i \neq t} L_i$$

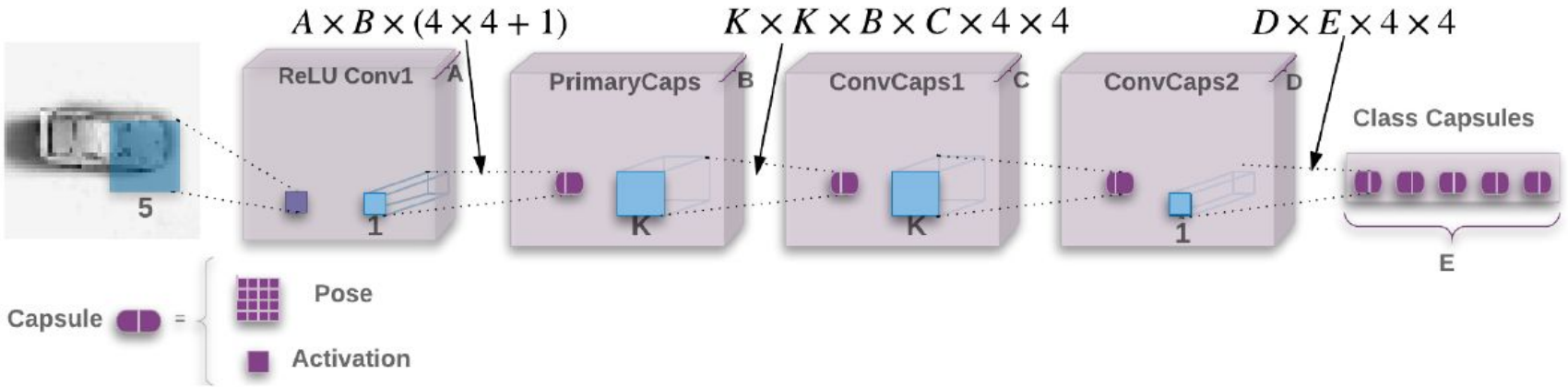
loss term for each class

margin. initial 0.2 and increased by 0.1 until 0.9

the activation for class i

Architecture

CapsNet with EMR



Datasets

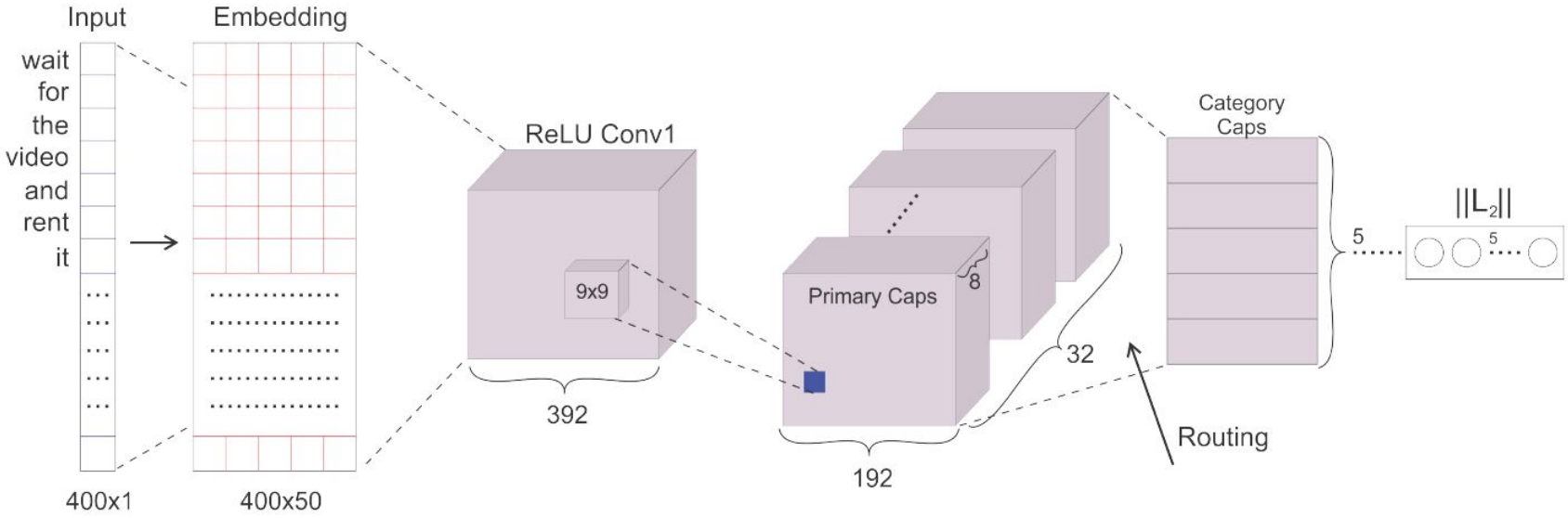
Approach

Data	Classes	ASL	DS	VS	TS
MR	2	10	10662	18008	CV
SST-1	5	10	11855	17471	2210
SST-2	2	10	9163	1741	1821
SUBJ	2	13	10000	20491	CV
TREC	6	5	5952	8328	500
ProcCons	2	6	45875	8972	CV
IMDB	2	120	50000	99455	CV

ASL - Average Sentence Length, DS - Dataset Size
VS - Vocabulary Size, TS - Test Size, CV - no standard train/test split

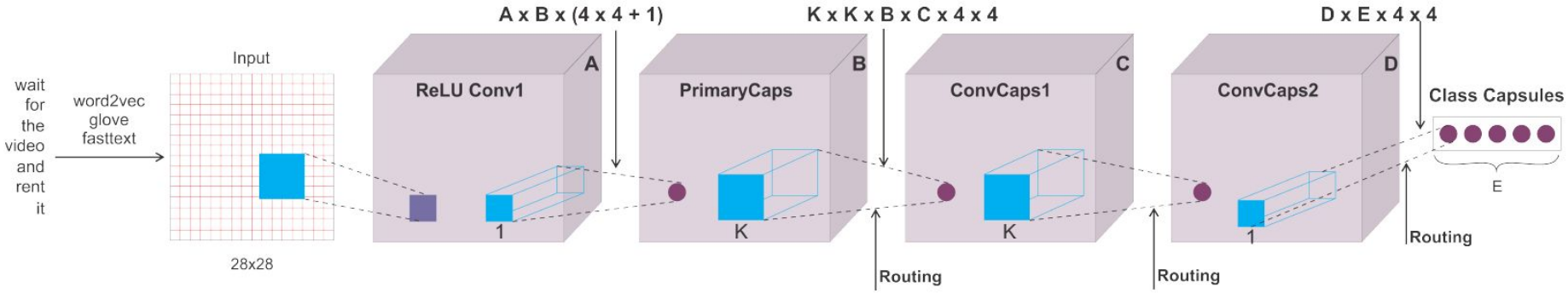
CapsNet with DR

Approach



CapsNet with EMR

Approach



Experiment 1: CapsNet with DR

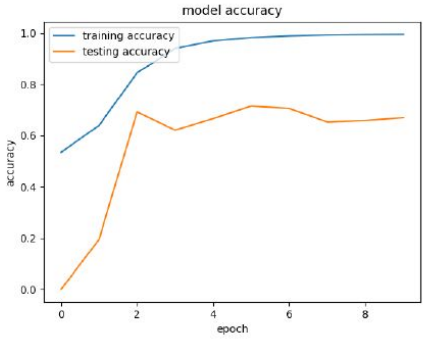
Evaluation

- Datasets: MR, SST-1, SST-2, SUBJ, TREC, ProcCons, IMDB
- Language: Python(Keras library)
- Dataset Split: Train: 70%, Dev: 15%, Test: 15%
- Optimizer: Adam, Nadam
- Epoch Number: 10, 20
- Batch Size: 200, 500
- Learning Rate Scheduler:
  - lambda1(epoch) : return 0.001 * exp(-epoch/10)
  - lambda2(epoch) : return 0.001/sqrt(epoch + epsilon)
  - step_decay(epoch) : 0.1*0.5^(floor(1+epoch)/5)

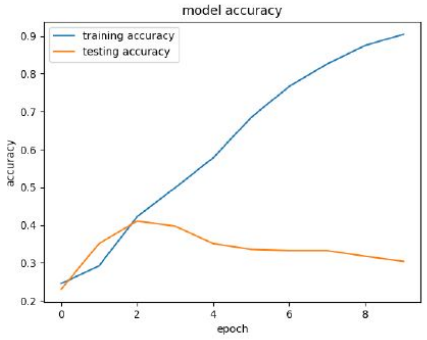
o	en	bz	lr	MR	SST-1	SST-2	SUBJ	TREC
a	10	200	l1	71%	34%	**73.8%**	87.9%	73%
a	10	200	l2	20%	11%	50%	18%	3%
a	10	200	sd	0%	30%	52.2%	33%	10%
a	10	500	l1	66%	22%	72.5%	**89%**	43%
a	10	500	l2	27%	16%	51%	56%	3%
a	10	500	sd	45%/99%	12%	51,7%	12%	29%
a	20	200	l1	70%	33%	69.9%	82%	76%
a	20	200	l2	30%	25%	50.4%	28%	3%
a	20	200	sd	0%	16%	48.2%	44%/99%	30%
a	20	500	l1	69%	32%	72.3%	86%	61%
a	20	500	l2	3%	11%	49.7%	7%	7%
a	20	500	sd	58%	16%	50%	25%	10%
na	10	200	l1	**73%**	**34%**	71.6%	87%	75.5%
na	10	200	l2	26%	11%	51.4%	21%	3%
na	10	200	sd	0%	22%	48.2%	0%	3%
na	10	500	l1	57%	34%	73%	87%	50%
na	10	500	l2	0%	13%	52.7%	45%/99%	12%
na	10	500	sd	44%/100%	16%	53%	45%/99%	3%
na	20	200	l1	71%	34%	70%	84%	**76.5%**
na	20	200	l2	2%	18%	53%	25%	3%
na	20	200	sd	44%/100%	18%	46%	45%/99%	3%
na	20	500	l1	61%	34%	67%	82%	66%
na	20	500	l2	1%	14%	50%	13%	18%
na	20	500	sd	45%/99%	22%	46%	4%	3%
				73%	**34%**	**73.8%**	**89%**	**76.5%**

o	en	bz	lr	ProcCons	IMDB
a	10	200	l1	90.95%	81%
a	10	200	l2	29%	15%
a	10	200	sd	45%/100%	44%
a	10	500	l1	91.63%	79%
a	10	500	l2	19.42%	25%
a	10	500	sd	40%	45%/100%
a	20	200	l1	90.19%	82%
a	20	200	l2	37.23%	38%
a	20	200	sd	63.01%	39%
a	20	500	l1	90.65%	78%
a	20	500	l2	40.98%	31%
a	20	500	sd	45%/100%	11%
na	10	200	l1	90.43%	80%
na	10	200	l2	1.55%	30%
na	10	200	sd	18.79%	34%
na	10	500	l1	90.41%	79%
na	10	500	l2	45%/100%	25%
na	10	500	sd	40%/100%	15%
na	20	200	l1	90.24%	80%
na	20	200	l2	13.54%	30%
na	20	200	sd	40%/100%	40%/100%
na	20	500	l1	90.61%	77%
na	20	500	l2	29.01%	26%
na	20	500	sd	45%/100%	73%
				91.63%	**82%**

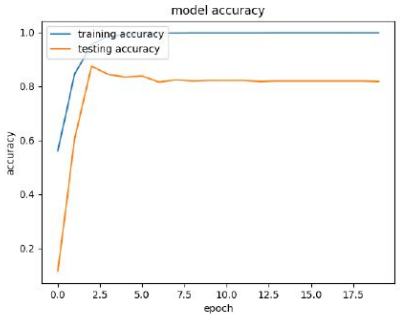
Accuracy



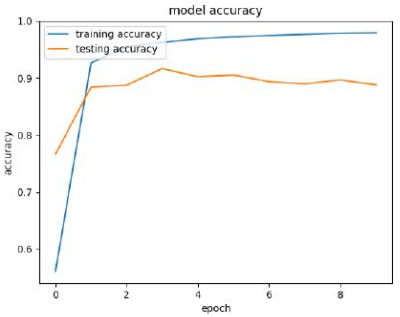
(a) MR



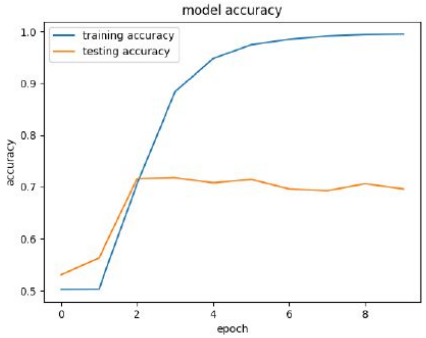
(b) SST-1



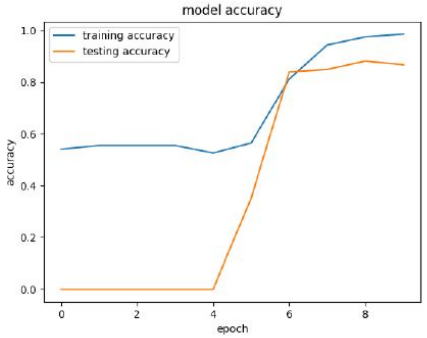
(e) TREC



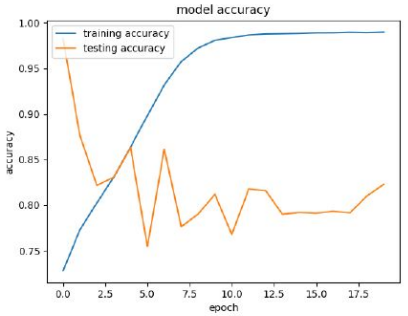
(f) ProcCons



(c) SST-2

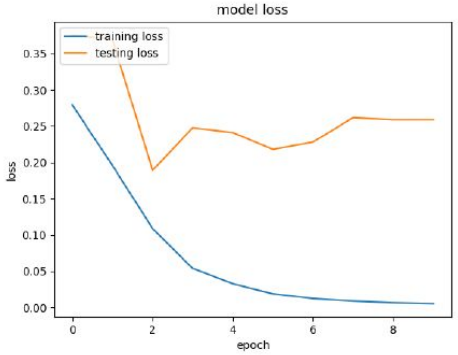


(d) SUBJ

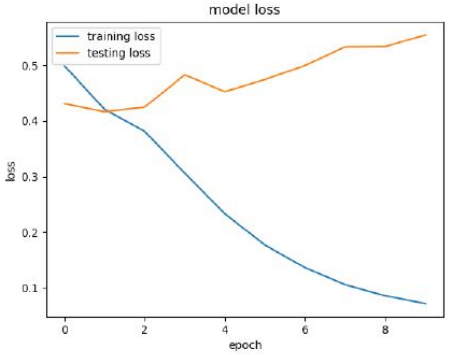


(g) IMDB

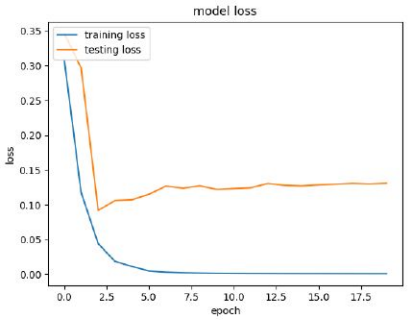
Loss



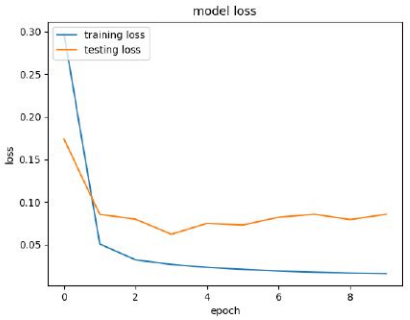
(a) MR



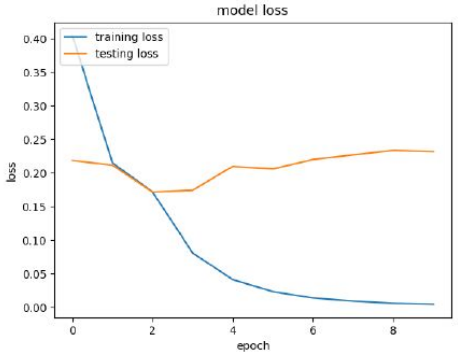
(b) SST-1



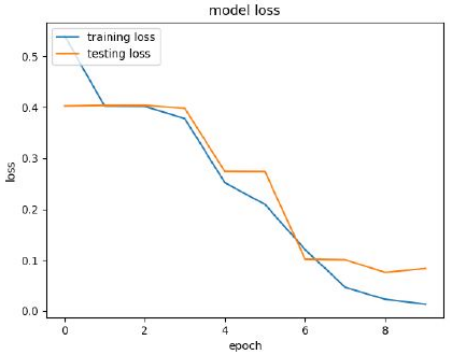
(e) TREC



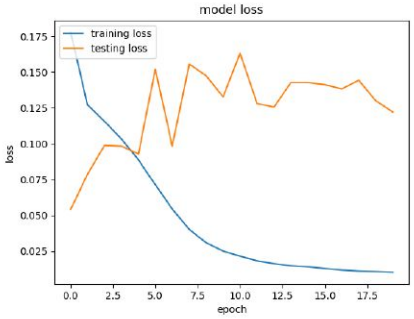
(f) ProcCons



(c) SST-2



(d) SUBJ



(g) IMDB

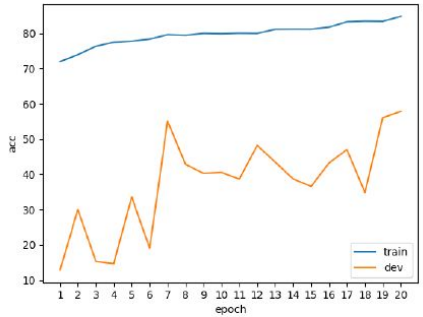
Evaluation

Experiment 2: CapsNet with EMR

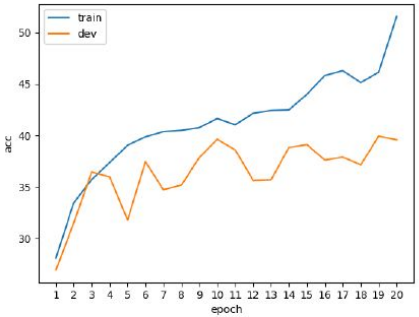
- Datasets: MR, SST-1, SST-2, SUBJ, TREC
- Language: Keray(PyTorch library)
- Optimizer: Adam, Adagrad
- Embeddings: Glove, Word2Vec, Fasttext
- PCA Dimension Reduction: 300D -> 28D
- Epoch Number: **20,** Batch Size: **64**
- Learning Rate Scheduler:
  - ***rp*** - ReduceLROnPlateau: decrease learning rate when a metric has stopped improving.
  - ***slr*** - StepLR: initialize the learning rate parameter with input lr and was reduced by gamma every step.

em	o	lr	MR	SST-1	SST-2	SUBJ	TREC
glove	a	rp	44.56%	39.17%	70.37%	78.66%	65.86%
glove	a	slr	43.00%	35.7%	69.38%	77.51%	65.74%
glove	ag	rp	44.12%	34.72%	66.72%	77.44%	65.62%
glove	ag	slr	47.50%	33.39%	66.49%	79.6%	64.9%
word2vec	a	rp	**57.87%**	**39.35%**	70.94%	75.88	59.61%
word2vec	a	slr	47.06%	38.25%	71.58%	74.04	61.89%
word2vec	ag	rp	49.43%	36.34%	71%	73.02%	60.57%
word2vec	ag	slr	50%	33.66%	70.08%	73.23%	59.61%
fasttext	a	rp	47.5%	35.99%	71.12%	**80.7%**	65.14%
fasttext	a	slr	47%	33.55%	**71.7%**	73.64%	**66.58%**
fasttext	ag	rp	47%	33.78%	70.25%	75.13%	64.9%
fasttext	ag	slr	49.56%	33.7%	69.38%	73.36%	62.98%
			57.87%	**39.35%**	**71.7%**	**80.7%**	**66.58%**

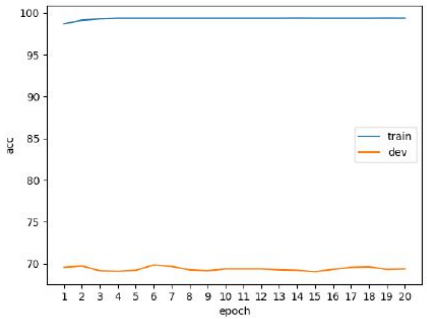
Accuracy



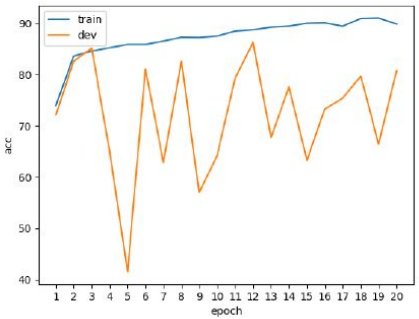
(a) MR



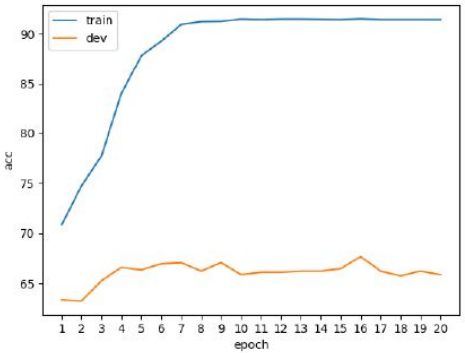
(b) SST-1



(c) SST-2

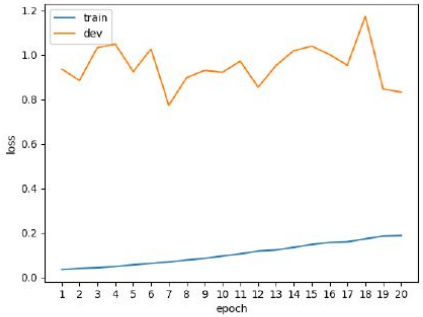


(d) SUBJ

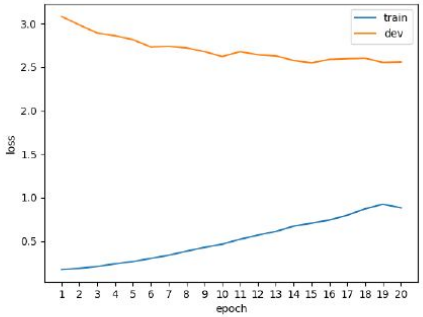


(e) TREC

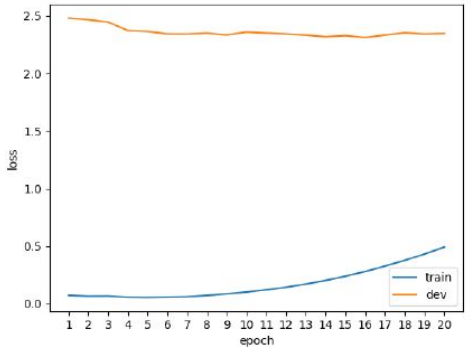
Loss



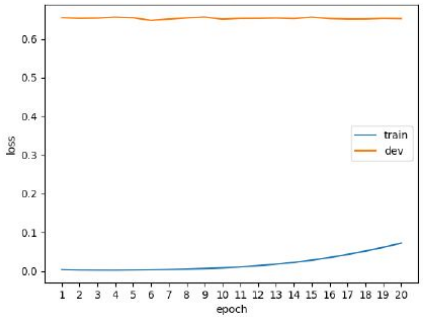
(a) MR



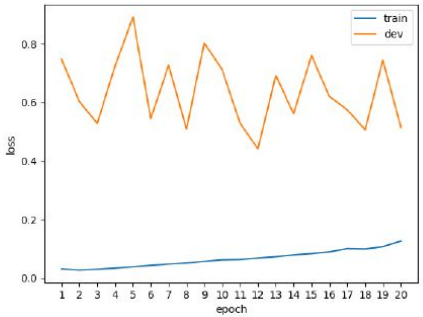
(b) SST-1



(e) TREC



(c) SST-2



(d) SUBJ

Comparison with state-of-the-art

Models	MR	SST-1	SST-2	SUBJ	TREC	ProcCons	IMDB
SA by Caps	**83.8%**	**49.3%**	-	-	-	-	-
CNN for SC	81.1%	47%	88.1%	**90%**	91.2%	-	-
CRDLM for SC	-	48%	**89.2%**	-	-	-	**93.4%**
CapsNet, DR	73%	34%	73.8%	89%	76.5%	**91.63%**	82%
CapsNet, EMR	57.87%	39.35%	71.7%	80.7%	66.58%	-	-

CapsNet with DR

Conclusion

- CapsNet with DR works better for big datasets
- Overfitting reasons:
  - Small datasets
  - Internal trainable embedding layer(sentence:400x50 tensor)
- Possible solutions:
  - Pre-trained word vectors: Glove, Word2Vec, Fasttext
  - Use 2D convolution operation
  - More data

CapsNet with EMR

Conclusion

- CapsNet with EMR too complex for small datasets
- Losing information:
  - Using PCA dimensional reducing
  - Pre-trained word to vectors do not have all words
  - Each sentence has a lot of empty words(zero-vectors)
- Possible solutions:
  - Possible use one low-level and one high-level capsule layer
  - Use dropout layer or L1/L2 (regularization methods)
  - More data

UNIVERSITÄT BONN

SMART DATA ANALYTICS
FROM DATA TO KNOWLEDGE

Fraunhofer IAIS

THANK YOU !