Supplementary Material of:
Do Neural Network Cross-Modal Mappings Really Bridge Modalities?

Guillem Collell and Marie-Francine Moens
Department of Computer Science
KU Leuven
gcollell@kuleuven.be ; sien.moens@cs.kuleuven.be

1 Hyperparameters and Implementation

Hyperparameters (including number of epochs) are chosen by 5 fold cross-validation (CV) optimizing for the test loss. Crucially, we ensure that all mappings are learned properly by verifying that the training loss steadily decreases. We search learning rates in {0.01, 0.001, 0.0001} and number of hidden units (d_h) in {64, 128, 256, 512, 1024}.

Using different number of hidden units (and selecting the best-performing one) is important in order to guarantee that our conclusions are not influenced or just a product of underfitting or overfitting. Similarly, we learned the mappings at different levels of dropout {0.25, 0.5, 0.75} which did not yield any improvement w.r.t. zero dropout (shown in our results).

We use a ReLu activation, the RMSprop optimizer ($\rho = 0.9$, $\epsilon = 10^{-8}$) and a batch size of 64. We find that sigmoid and tanh yield similar results as ReLu. Our implementation is in Keras (Chollet et al., 2015).

Since ImageNet does not have any set of "test concepts", we employ 5-fold CV. Reported results are either averages on 5 folds (ImageNet) or 5 runs with different model weights initializations (IAPR TC-12 and Wiki).

For the *max-margin* loss, we choose the margin γ by cross-validation and explore values within $\{1, 2.5, 5, 7.5, 10\}$.

2 Textual Feature Extraction

Unlike ImageNet where we associate a word embedding to each concept, the textual modality in IAPR TC-12 and Wiki consists of sentences. In order to extract state-of-the art textual features in these datasets we train the following, separate network (prior to the cross-modal mapping). First, the embedded input sentences are passed to a bidirectional GRU of 64 units, then fed into a fully-connected layer, followed by a cross-entropy loss on the vector of class labels. We collect the 64-d averaged GRU hidden states of both directions as features. The network is trained with the Adam optimizer.

In Wiki and IAPR TC-12 we verify that the extracted text and image features are indeed informative and useful by computing their mean average precision (mAP) in retrieval (considering that a document B is relevant for document A if A and B share at least one class label). In Wiki we find mAPs of: biGRU = 0.77, ResNet = 0.22 and vgg128 = 0.21. In IAPR TC-12 we find mAPs of: biGRU = 0.77, ResNet = 0.49 and vgg128 = 0.46. Notice that ImageNet has a single data point per class in our setting, and thus mAP cannot be computed. However, we employ standard GloVe, word2vec, VGG-128 and ResNet vectors in ImageNet, which are known to perform well.

3 Additional Results

Results with *mNNO*(X, Y) (omitted in the main paper for space reasons): Interestingly, the similarity *mNNO*(X, Y) between original input X and output Y vectors is generally low (between 1.5 and 2.3), indicating that these spaces are originally quite different. However, *mNNO*(X, Y) always remains lower than *mNNO*$(f(X), Y)$, indicating thus that the mapping makes a difference.

3.1 Experiment 1

3.1.1 Results with 3 and 5 layers

			ResNet		VGG-128	
			$X, f(X)$	$Y, f(X)$	$X, f(X)$	$Y, f(X)$
ImageNet	$I \to T$	nn-3	**0.571**	0.279	**0.602**	0.258
		nn-5	**0.615**	0.275	**0.644**	0.255
	$T \to I$	nn-3	**0.274**	0.27	0.254	**0.256**
		nn-3	**0.286**	0.274	**0.273**	0.259
IAPR TC	$I \to T$	nn-5	**0.301**	0.225	**0.288**	0.181
		nn-3	**0.29**	0.227	**0.308**	0.184
	$T \to I$	nn-3	**0.324**	0.229	**0.294**	0.18
		nn-5	**0.355**	0.232	**0.339**	0.183
Wiki	$I \to T$	nn-3	**0.227**	0.159	**0.247**	0.144
		nn-5	**0.275**	0.163	**0.262**	0.146
	$T \to I$	nn-3	**0.367**	0.148	**0.342**	0.145
		nn-5	**0.412**	0.152	**0.428**	0.147

Table 1: Test mean nearest neighbor overlap with 3- and 5-hidden layer neural networks, using cosine-based neighbors and MSE loss. Boldface indicates best performance between each *mNNO*$^{10}(X, f(X))$ and *mNNO*$^{10}(Y, f(X))$ pair, which are abbreviated by $X, f(X)$ and $Y, f(X)$.

It is interesting to notice that even though the difference between *mNNO*$^{10}(X, f(X))$ and *mNNO*$^{10}(Y, f(X))$ has narrowed down w.r.t. the linear and 1-hidden layer models (in the main paper) in some cases (e.g., ImageNet), this does not seem to be caused by better predictions, i.e., an increase of *mNNO*$^{10}(Y, f(X))$, but rather by a decrease of *mNNO*$^{10}(X, f(X))$. This is expected since with more layers the information about the input is less preserved.

			ResNet		VGG-128	
			$X, f(X)$	$Y, f(X)$	$X, f(X)$	$Y, f(X)$
ImageNet	$I \to T$	nn-3	**0.562**	0.243	**0.574**	0.229
		nn-5	**0.61**	0.241	**0.619**	0.227
	$T \to I$	nn-3	0.252	**0.263**	0.23	**0.244**
		nn-3	0.261	**0.264**	**0.243**	0.242
IAPR TC	$I \to T$	nn-5	**0.275**	0.208	**0.259**	0.174
		nn-3	**0.262**	0.207	**0.276**	0.174
	$T \to I$	nn-3	**0.312**	0.215	**0.27**	0.168
		nn-5	**0.351**	0.218	**0.315**	0.17
Wiki	$I \to T$	nn-3	**0.219**	0.15	**0.239**	0.14
		nn-5	**0.259**	0.152	**0.25**	0.143
	$T \to I$	nn-3	**0.375**	0.145	**0.363**	0.134
		nn-5	**0.431**	0.144	**0.426**	0.135

Table 2: Test mean nearest neighbor overlap with 3- and 5-hidden layer neural networks, using Euclidean neighbors and MSE loss.

3.1.2 Results with the max margin loss

			ResNet		VGG-128	
			$X, f(X)$	$Y, f(X)$	$X, f(X)$	$Y, f(X)$
ImageNet	$I \to T$	lin	**0.739**	0.253	**0.779**	0.235
		nn	**0.769**	0.233	**0.736**	0.238
	$T \to I$	lin	**0.526**	0.252	**0.454**	0.241
		nn	**0.419**	0.23	**0.378**	0.22
IAPR TC	$I \to T$	lin	**0.423**	0.205	**0.441**	0.164
		nn	**0.291**	0.179	**0.36**	0.16
	$T \to I$	lin	**0.674**	0.198	**0.604**	0.17
		nn	**0.592**	0.215	**0.529**	0.176
Wiki	$I \to T$	lin	**0.366**	0.156	**0.333**	0.152
		nn	**0.236**	0.153	**0.399**	0.153
	$T \to I$	lin	**0.725**	0.151	**0.723**	0.146
		nn	**0.701**	0.151	**0.662**	0.146

Table 3: Test mean nearest neighbor overlap with cosine-based neighbors and *max margin loss*.

			ResNet		VGG-128	
			$X, f(X)$	$Y, f(X)$	$X, f(X)$	$Y, f(X)$
ImageNet	$I \to T$	lin	**0.762**	0.229	**0.776**	0.209
		nn	**0.776**	0.213	**0.724**	0.214
	$T \to I$	lin	**0.49**	0.241	**0.418**	0.225
		nn	**0.384**	0.221	**0.343**	0.212
IAPR TC	$I \to T$	lin	**0.409**	0.195	**0.447**	0.155
		nn	**0.275**	0.172	**0.329**	0.15
	$T \to I$	lin	**0.685**	0.189	**0.619**	0.158
		nn	**0.558**	0.201	**0.49**	0.162
Wiki	$I \to T$	lin	**0.38**	0.154	**0.339**	0.142
		nn	**0.232**	0.144	**0.398**	0.141
	$T \to I$	lin	**0.789**	0.143	**0.773**	0.135
		nn	**0.724**	0.14	**0.723**	0.135

Table 4: Test mean nearest neighbor overlap with Euclidean-based neighbors and *max margin loss*.

3.1.3 Results with the cosine loss

			ResNet		VGG-128	
			$X, f(X)$	$Y, f(X)$	$X, f(X)$	$Y, f(X)$
ImageNet	$I \to T$	lin	**0.697**	0.268	**0.812**	0.244
		nn	**0.58**	0.28	**0.629**	0.256
	$T \to I$	lin	**0.382**	0.241	**0.336**	0.224
		nn	**0.346**	0.277	**0.331**	0.237
IAPR TC	$I \to T$	lin	**0.37**	0.213	**0.594**	0.162
		nn	**0.35**	0.234	**0.516**	0.158
	$T \to I$	lin	**0.469**	0.205	**0.405**	0.169
		nn	**0.386**	0.226	**0.338**	0.185
Wiki	$I \to T$	lin	**0.26**	0.157	**0.621**	0.143
		nn	**0.213**	0.156	**0.281**	0.15
	$T \to I$	lin	**0.549**	0.157	**0.53**	0.154
		nn	**0.642**	0.151	**0.547**	0.149

Table 5: Test mean nearest neighbor overlap with cosine-based neighbors and *cosine loss*.

			ResNet		VGG-128	
			$X, f(X)$	$Y, f(X)$	$X, f(X)$	$Y, f(X)$
ImageNet	$I \to T$	lin	**0.698**	0.236	**0.812**	0.218
		nn	**0.562**	0.238	**0.597**	0.218
	$T \to I$	lin	**0.36**	0.225	**0.319**	0.209
		nn	**0.28**	0.221	**0.288**	0.205
IAPR TC	$I \to T$	lin	**0.351**	0.197	**0.596**	0.152
		nn	**0.295**	0.201	**0.452**	0.144
	$T \to I$	lin	**0.475**	0.184	**0.409**	0.153
		nn	**0.359**	0.193	**0.29**	0.158
Wiki	$I \to T$	lin	**0.259**	0.149	**0.619**	0.133
		nn	**0.212**	0.147	**0.262**	0.144
	$T \to I$	lin	**0.527**	0.147	**0.496**	0.137
		nn	**0.578**	0.143	**0.51**	0.135

Table 6: Test mean nearest neighbor overlap with Euclidean-based neighbors and *cosine loss*.

3.1.4 Results with Euclidean Neighbors (*nn* and *lin* models of the paper)

			ResNet		VGG-128	
			$X, f(X)$	$Y, f(X)$	$X, f(X)$	$Y, f(X)$
ImageNet	$I \to T$	lin	**0.671**	0.228	**0.695**	0.209
		nn	**0.61**	0.234	**0.665**	0.219
	$T \to I$	lin	**0.372**	0.233	**0.326**	0.218
		nn	**0.332**	0.258	**0.298**	0.242
IAPR TC	$I \to T$	lin	**0.341**	0.194	**0.385**	0.156
		nn	**0.3**	0.203	**0.318**	0.17
	$T \to I$	lin	**0.504**	0.188	**0.431**	0.156
		nn	**0.421**	0.21	**0.363**	0.169
Wiki	$I \to T$	lin	**0.245**	0.146	**0.235**	0.141
		nn	**0.261**	0.151	**0.269**	0.143
	$T \to I$	lin	**0.564**	0.149	**0.555**	0.135
		nn	**0.539**	0.149	**0.529**	0.14

Table 7: Test mean nearest neighbor overlap with Euclidean-based neighbors and MSE loss. Boldface indicates best performance between each *mNNO*$^{10}(X, f(X))$ and *mNNO*$^{10}(Y, f(X))$ pair, which are abbreviated by $X, f(X)$ and $Y, f(X)$.

		ResNet		VGG-128	
		$X, f(X)$	$Y, f(X)$	$X, f(X)$	$Y, f(X)$
$I \to T$	lin	**0.57**	0.16	**0.644**	0.159
	nn	**0.546**	0.179	**0.64**	0.171
$T \to I$	lin	**0.325**	0.206	**0.283**	0.2
	nn	**0.283**	0.236	**0.259**	0.223

Table 8: Test *mNNO* with Euclidean-based neighbors in **ImageNet** dataset, using *word2vec* word embeddings.

3.1.5 Results with word2vec in ImageNet (cosine-based neighbors)

		ResNet		VGG-128	
		$X, f(X)$	$Y, f(X)$	$X, f(X)$	$Y, f(X)$
$I \to T$	lin	**0.61**	0.232	**0.674**	0.221
	nn	**0.578**	0.253	**0.666**	0.236
$T \to I$	lin	**0.364**	0.213	**0.348**	0.21
	nn	**0.356**	0.245	**0.331**	0.234

Table 9: Test *mNNO* using cosine-based neighbors in **ImageNet**, using *word2vec* word embeddings.

3.2 Experiment 2

	WS-353		Men		SemSim	
	Cos	Eucl	Cos	Eucl	Cos	Eucl
f_{nn}(word2vec)	0.665	0.636	0.782	0.781	0.729	0.719
f_{lin}(word2vec)	0.67	0.527	0.785	0.696	0.737	0.616
word2vec	0.669	0.533	0.787	0.701	0.742	0.62
f_{nn}(VGG-128)	0.44	0.433	0.588	0.585	0.521	0.513
f_{lin}(VGG-128)	0.445	0.301	0.593	0.496	0.531	0.344
VGG-128	0.448	0.307	0.593	0.496	0.534	0.344

	VisSim		SimLex		SimVerb	
	Cos	Eucl	Cos	Eucl	Cos	Eucl
f_{nn}(word2vec)	0.566	0.567	0.419	0.379	0.309	0.232
f_{lin}(word2vec)	0.572	0.507	0.429	0.275	0.328	0.174
word2vec	0.576	0.51	0.435	0.279	0.308	0.15
f_{nn}(VGG-128)	0.551	0.547	0.404	0.399	0.231	0.235
f_{lin}(VGG-128)	0.56	0.404	0.406	0.335	0.23	0.316
VGG-128	0.56	0.403	0.406	0.335	0.235	0.329

Table 10: Spearman correlations between human ratings and similarities (cosine or Euclidean) predicted from the embeddings, using *word2vec* and *VGG-128* embeddings.

References

François Chollet et al. 2015. Keras. `https://github.com/keras-team/keras`.