Zero-Shot Learning by Convex Combination of Semantic Embeddings

Mohammad Norouzi , Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S. Corrado and Jeffrey Dean

University of Toronto Google, Inc. ON, Canada Mountain View, CA, USA

Presented by: Youmna Farag

Zero-Shot Learning

- Problem: Annotating large number of object categories is challenging and expensive and needs updating over time to include new objects.
- Zero-Shot Learning: “The ability to correctly annotate images of previously unseen object categories”
- Solution: Mapping images into semantic embedding spaces. (trying to find relationships between object categories)

Training images $\mathcal{Y}_0$

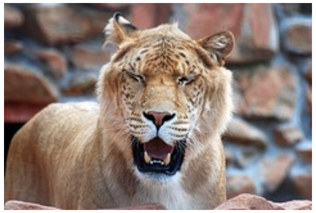
=> Tiger

=> Lion

=> Rat

=> Fish

Test images $\mathcal{Y}_1$



=> Liger

=> Dog

Semantic Embedding Approaches

- Attribute Based Approaches
  - E.g. Binary attributes to encode presence or absence of attributes in object, such as materials, colors and object parts.
  - Disadvantages: Scalability issue, the need to annotate thousands of classes with thousands of attributes
- Unsupervised Neural Language Modeling
  - Learn a set of embedding vectors for words in a corpus, use that to embed class labels

Problem Statement

- Training datas $\mathcal{D}_0 \equiv \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$ $\mathbf{x}_i \in \mathbb{R}^p$ e is the fe $y_i \in \mathcal{Y}_0 \equiv \{1, \ldots, n_0\}$ are training labels
- Test datase $\mathcal{D}_1 \equiv \{(\mathbf{x}'_j, y'_j)\}_{j=1}^{m'}$ where $y'_j \in \mathcal{Y}_1 \equiv$ $\{n_0+1, \ldots, n_0+n_1\}$ are test labels
- $\mathcal{Y}_0 \cap \mathcal{Y}_1 = \varnothing$
- Associate all labels with semantic embedding vector
- $\mathbf{s}(y) \in \mathcal{S} \equiv \mathbb{R}^q$ so that $\{\mathbf{s}(y);\, y \in \mathcal{Y}_0 \cup \mathcal{Y}_1\}$
- *y* is similar to *y'* if *s(y)* is close to *s(y')*

Regression Model

- Map input features to semantic embedding vectors using a regre$(\mathcal{X} \to \mathcal{S})$nodel ,
  instead of lear$(\mathcal{X} \to \mathcal{Y}_0)$-way classifier
- Training se$\{(\mathbf{x}_i, s(y_i));\ (\mathbf{x}_i, y_i) \in \mathcal{D}_0\}$
- Learn a regression function $f : \mathcal{X} \to \mathcal{S}$
- Use k-nearest neighbor search in the semantic space to map the prediction in $\mathcal{S}$ to a ranked list of labels in $\mathcal{Y}_1$

Convex Combination of Semantic Embeddings (ConSE)

- Learn a classifier *p0* to map training inputs to labels
- Output is a set of probabi $p_0(y \mid \mathbf{x})$ for class labels $\sum_{y=1}^{n_0} p_0(y \mid \mathbf{x}) = 1$.
- $\widehat{y}_0(\mathbf{x}, 1)$ is the most likely training label for image *x*:

$$\widehat{y}_0(\mathbf{x}, 1) \equiv \underset{y \in \mathcal{Y}_0}{\operatorname{argmax}}\, p_0(y \mid \mathbf{x})$$

- Similarly, is the tth most likely label
- Given to $\widehat{y}_0(\mathbf{x}, t)$ dictions, predict a semantic embedding *f(x)* as the convex combination of the semantic embedding $\{s(\widehat{y}_0(\mathbf{x}, t))\}_{t=1}^{T}$ d by their probabilities

$$f(\mathbf{x}) = \frac{1}{Z} \sum_{t=1}^{T} p(\widehat{y}_0(\mathbf{x}, t) \mid \mathbf{x}) \cdot s(\widehat{y}_0(\mathbf{x}, t))$$ where $Z = \sum_{t=1}^{T} p(\widehat{y}_0(\mathbf{x}, t) \mid \mathbf{x})$

Convex Combination of Semantic Embeddings (ConSE)

- $$f(\mathbf{x}) = \frac{1}{Z} \sum_{t=1}^{T} p(\widehat{y}_0(\mathbf{x}, t) \mid \mathbf{x}) \cdot s(\widehat{y}_0(\mathbf{x}, t))$$
- Example: *p0*=(lion|*x*) = 0.6 and *p0*=(tiger|*x*) = 0.4, *f(x)* = 0.6 . *s*(lion) + 0.4 . *s*(tiger). Giving “liger”, a hybrid between lion and tiger. f(x) ≈ s(liger)
- For prediction: find test labels with embeddings neares f image *x* is calcul

$$\widehat{y}_1(\mathbf{x}, 1) \equiv \operatorname*{argmax}_{y' \in \mathcal{Y}_1} cos(f(\mathbf{x}), s(y'))$$

- $\widehat{y}_1(\mathbf{x}, k)$ is the label with the *kth* largest value of cosine similarity

Models

- Softmax Baseline (krizhevsky et al. 2012):

deep convolutional neural network (CNN) to classify images from ImageNet. Can only predict the labels seen in training data.

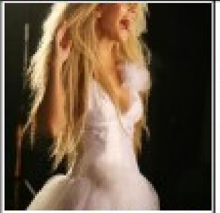
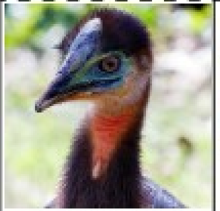
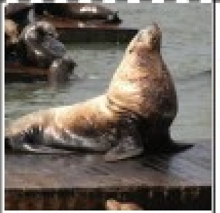
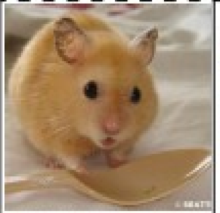
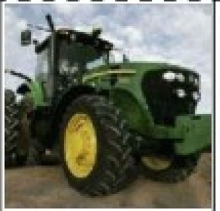
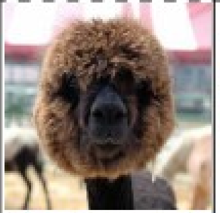
- Deep Visual-Semantic Embedding (DeViSE) (Frome et al. 2013):
  - Use same CNN in krizhevsky et al.
  - Use skip-gram model to generate the semantic embedding space
  - Replace softmax layer with a linear transformation layer
  - Transformation layer is trained using a ranking objective to map training inputs to embedding vectors close to correct labels
- ConSE:
  - Use same CNN in krizhevsky et al., keeping the softmax layer

Data

- Semantic embedding space:

skip-gram model trained on 5.4 billion words from Wikipedia.org to construct 500 dimensional word embedding vectors

- Images:
  - Training: ImageNet 2012 1K set with 1000 training labels
  - Test:
  - “2-hops”: labels from the 2011 21K set which are visually and semantically similar to the training labels (labels within 2 tree hops) – size = 1,589
  - “3-hops”: labels from the 2011 21K set within 3 tree hops training labels. (a more difficult set) –

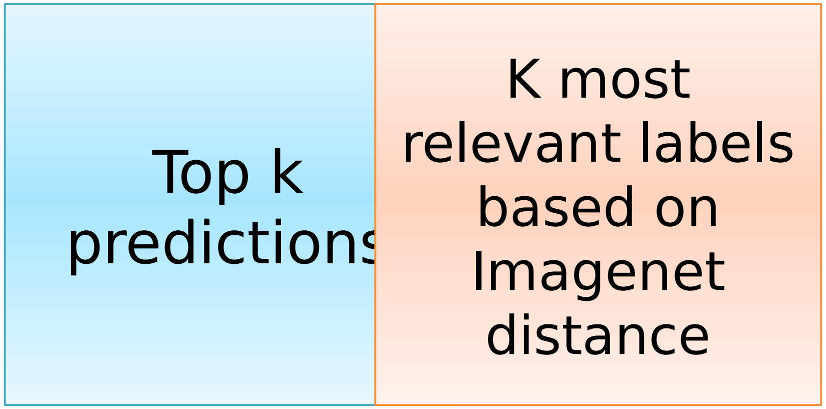
Test Image	Softmax Baseline [7]	DeViSE [6]	ConSE (10)
	wig fur coat Saluki, gazelle hound Afghan hound, Afghan stole	water spaniel tea gown bridal gown, wedding gown spaniel tights, leotards	business suit **dress, frock** hairpiece, false hair, postiche swimsuit, swimwear, bathing suit kit, outfit
	ostrich, Struthio camelus black stork, Ciconia nigra vulture crane peacock	heron owl, bird of Minerva, bird of night hawk bird of prey, raptor, raptorial bird finch	**ratite, ratite bird, flightless bird** peafowl, bird of Juno common spoonbill New World vulture, cathartid Greek partridge, rock partridge
	sea lion plane, carpenter's plane cowboy boot loggerhead, loggerhead turtle goose	elephant turtle turtleneck, turtle, polo-neck flip-flop, thong handcart, pushcart, cart, go-cart	California sea lion **Steller sea lion** Australian sea lion South American sea lion eared seal
	hamster broccoli Pomeranian capuchin, ringtail weasel	**golden hamster, Syrian hamster** rhesus, rhesus monkey pipe shaker American mink, Mustela vison	**golden hamster, Syrian hamster** rodent, gnawer Eurasian hamster rhesus, rhesus monkey rabbit, coney, cony
 **(farm machine)**	thresher, threshing machine tractor harvester, reaper half track snowplow, snowplough	truck, motortruck skidder tank car, tank automatic rifle, machine rifle trailer, house trailer	flatcar, flatbed, flat truck, motortruck tracked vehicle bulldozer, dozer wheeled vehicle
 **(alpaca, Lama pacos)**	Tibetan mastiff titi, titi monkey koala, koala bear, kangaroo bear llama chow, chow chow	kernel littoral, litoral, littoral zone, sands carillon Cabernet, Cabernet Sauvignon poodle, poodle dog	dog, domestic dog domestic cat, house cat schnauzer Belgian sheepdog domestic llama, Lama peruana

Evaluation

- “flat” hit@*k*:
  - the percentage of test images for which the model returns the one true label in its top k predictions.
- “hierarchical” precision@*k:*
  - uses the ImageNet category hierarchy to penalize the predictions that are semantically far from the correct labels more than the predictions that are close.

Evaluation

- “flat” hit@*k*:
  - the percentage of test images for which the model returns the one true label in its top k predictions.
- “hierarchical” precision@*k:*
  - uses the ImageNet category hierarchy to penalize the predictions that are semantically far from the correct labels more than the predictions that are close.

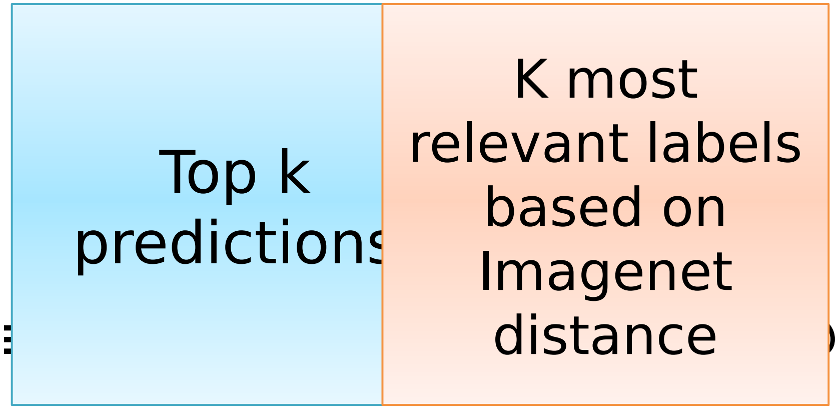


Top k predictions

K most relevant labels based on Imagenet distance

Evaluation

- “flat” hit@*k*:
  - the percentage of test images for which the model returns the one true label in its top k predictions.
- “hierarchical” precision@*k:*
  - uses the ImageNet category hierarchy to penalize the predictions that are semantically far from the correct labels more than the predictions that are close.



Top k predictions

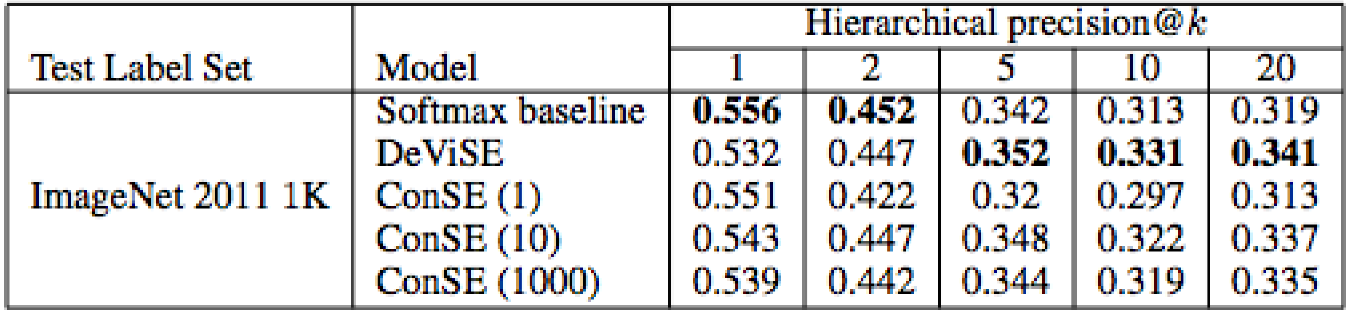
K most relevant labels based on Imagenet distance

- flat hit@*1* = 1

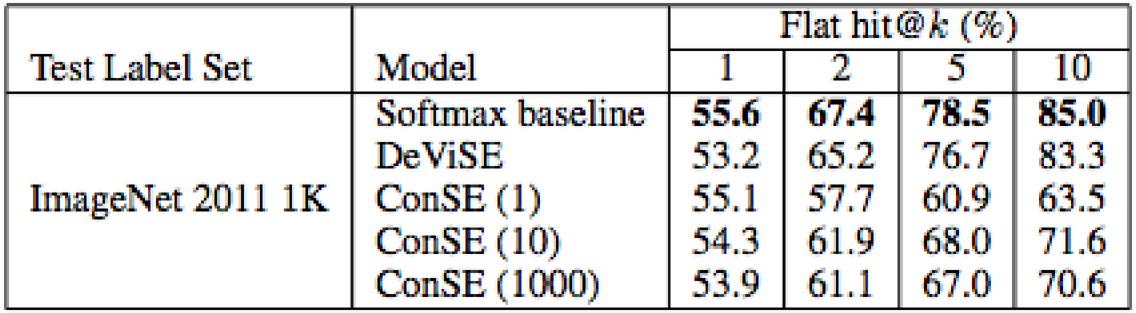
Test Label Set	# Candidate Labels	Model	Flat hit@k (%)				
			1	2	5	10	20
2-hops	1,589	DeViSE	6.0	10.0	18.1	26.4	36.4
		ConSE(1)	9.3	14.4	23.7	30.8	38.7
		ConSE(10)	**9.4**	**15.1**	**24.7**	**32.7**	**41.8**
		ConSE(1000)	9.2	14.8	24.1	32.1	41.1
2-hops (+1K)	1,589 +1000	DeViSE	**0.8**	2.7	7.9	14.2	22.7
		ConSE(1)	0.2	**7.1**	**17.2**	24.0	31.8
		ConSE(10)	0.3	6.2	17.0	**24.9**	**33.5**
		ConSE(1000)	0.3	6.2	16.7	24.5	32.9
3-hops	7,860	DeViSE	1.7	2.9	5.3	8.2	12.5
		ConSE(1)	2.6	4.2	7.3	10.8	14.8
		ConSE(10)	**2.7**	**4.4**	**7.8**	**11.5**	**16.1**
		ConSE(1000)	2.6	4.3	7.6	11.3	15.7
3-hops (+1K)	7,860 +1000	DeViSE	**0.5**	1.4	3.4	5.9	9.7
		ConSE(1)	0.2	**2.4**	**5.9**	9.3	13.4
		ConSE(10)	0.2	2.2	**5.9**	**9.7**	**14.3**
		ConSE(1000)	0.2	2.2	5.8	9.5	14.0
ImageNet 2011 21K	20,841	DeViSE	0.8	1.4	2.5	3.9	6.0
		ConSE(1)	1.3	2.1	3.6	5.4	7.6
		ConSE(10)	**1.4**	**2.2**	**3.9**	**5.8**	**8.3**
		ConSE(1000)	1.3	2.1	3.8	5.6	8.1
ImageNet 2011 21K (+1K)	20,841 +1000	DeViSE	**0.3**	0.8	1.9	3.2	5.3
		ConSE(1)	0.1	1.2	3.0	4.8	7.0
		ConSE(10)	0.2	1.2	3.0	**5.0**	**7.5**
		ConSE(1000)	0.2	1.2	3.0	4.9	7.3

Test Label Set	Model	Hierarchical precision@*k*				
		1	2	5	10	20
2-hops	DeViSE	0.06	0.152	0.192	0.217	0.233
	ConSE(10)	**0.094**	**0.214**	**0.247**	**0.269**	**0.284**
2-hops (+1K)	Softmax baseline	0	**0.236**	0.181	0.174	0.179
	DeViSE	**0.008**	0.204	0.196	0.201	0.214
	ConSE(10)	0.003	0.234	**0.254**	**0.260**	**0.271**
3-hops	DeViSE	0.017	0.037	0.191	0.214	0.236
	ConSE(10)	**0.027**	**0.053**	**0.202**	**0.224**	**0.247**
3-hops (+1K)	Softmax baseline	0	0.053	0.157	0.143	0.130
	DeViSE	**0.005**	0.053	0.192	0.201	0.214
	ConSE(10)	0.002	**0.061**	**0.211**	**0.225**	**0.240**
ImageNet 2011 21K	DeViSE	0.008	0.017	0.072	0.085	0.096
	ConSE(10)	**0.014**	**0.025**	**0.078**	**0.092**	**0.104**
ImageNet 2011 21K (+1K)	Softmax baseline	0	0.023	0.071	0.069	0.065
	DeViSE	**0.003**	0.025	0.083	0.092	0.101
	ConSE(10)	0.002	**0.029**	**0.086**	**0.097**	**0.105**

Training and Test Labels are the Same (no Zero-Shot Learning)



Test Label Set	Model	Hierarchical precision@k				
		1	2	5	10	20
ImageNet 2011 1K	Softmax baseline	**0.556**	**0.452**	0.342	0.313	0.319
	DeViSE	0.532	0.447	**0.352**	**0.331**	**0.341**
	ConSE (1)	0.551	0.422	0.32	0.297	0.313
	ConSE (10)	0.543	0.447	0.348	0.322	0.337
	ConSE (1000)	0.539	0.442	0.344	0.319	0.335



Test Label Set	Model	Flat hit@k (%)			
		1	2	5	10
ImageNet 2011 1K	Softmax baseline	**55.6**	**67.4**	**78.5**	**85.0**
	DeViSE	53.2	65.2	76.7	83.3
	ConSE (1)	55.1	57.7	60.9	63.5
	ConSE (10)	54.3	61.9	68.0	71.6
	ConSE (1000)	53.9	61.1	67.0	70.6

Implementation Details

- ConSE(1) occasionally differs from Softmax baseline prediction because:
  - There is no one-to-one correspondence between labels and embedding vectors
  - To softmax scores to embedding vectors, ConSE averages word vectors associated with each label (to mirror Imagenet synsets), then average vectors are linearly combined according to softmax scores.
  - i.e. this model takes synonym words into account

Conclusion

- ConSE is a simple model to map images to semantic embedding vectors
- ConSE outperforms other zero-short-learning approaches
- ConSE can use any other visual object classification system or text vector representations.
- ConSE can represent the system confidence
  - Labels of low probabilities reduces the

Conclusion

- ConSE is a simple model to map images to semantic embedding vectors
- ConSE outperforms other zero-short-learning approaches
- ConSE can use any other visual object classification system or text vector representations.
- ConSE can represent the system confid

$$f(\mathbf{x}) = \frac{1}{Z} \sum_{t=1}^{T} p(\widehat{y}_0(\mathbf{x}, t) \mid \mathbf{x}) \cdot s(\widehat{y}_0(\mathbf{x}, t))$$

  - –Labels of low probabilities reduces the

Thank You! □
Questions?