Machine Learning Reading Notes

Brendan Duke

April 2, 2017

1 Definitions

Deep Neural Networks (DNNs) are engineered systems inspired by the biological brain [1].

The **softmax function** is a continuous differentiable version of the argmax function, where the result is represented as a one-hot vector [1, Chapter 6]. Softmax is a way of representing probability distributions over a discrete variable that can take on n possible values.

Formally, softmax is given by Equation 1.

$$\operatorname{softmax}(\boldsymbol{z})_i = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{1}$$

Mahalanobis Distance

Neighbourhood Components Analysis (NCA) is a method of learning a Mahalanobis distance metric, and can also be used in linear dimensionality reduction [2].

The **PCKh** metric, used by the MPII Human Pose Dataset, defines a joint estimate as matching the ground truth if the estimate lies within 50% of the head segment length [3]. The head segment length is defined as the diagonal across the annotated head rectangle in the MPII data, multiplied by a factor of 0.6. Details can be found by examining the MATLAB evaluation script provided with the MPII dataset.

Non-maximum suppression in object detection, in general, is a set of methods used to prune an initial set of object bounding boxes that may be uncorrelated with the actual object detections in an image, down to a subset that are [4]. In edge detection, non-maximum suppression is used to suppress any pixels (i.e. not include them in the set of detected edges) that are not the maximum response in their neighbourhood.

LSTM (Long Short Term Memory) neural networks are a type of recurrent neural network whose characteristic feature is the presence of a gated self-loop that allows retention of its “cell state”, which are the pre-non-linearity activations of the previous time step [1, Chapter 10].

Cell state is updated at each time step according to Equation 2.

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma \left(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{(t-1)} \right) \tag{2}$$

The vectors $\boldsymbol{f}^{(t)}$ and $\boldsymbol{g}^{(t)}$ in Equation 2 also take inputs from $\boldsymbol{x}^{(t)}$ and $\boldsymbol{h}^{(t-1)}$, with their own weight tensors and bias vectors $\boldsymbol{U}^f$, $\boldsymbol{W}^f$ and $\boldsymbol{b}^f$, $\boldsymbol{U}^g$, $\boldsymbol{W}^g$ and $\boldsymbol{b}^f$, respectively.

Similar gate functions exist to gate the inputs and outputs to the LSTM, as well.

2 Paper Summaries

2.1 DeepPose: Human Pose Estimation via Deep Neural Networks [5]

This paper uses DNNs as a method for human pose estimation, based on the success of [6] and [7] for object detection using DNNs.

This is in contrast to the existing work in human pose estimation at the time, which focused on explicitly designed pose models. Papers about these methods can be found in the "Related Work" section of [5].

The input to the 7-layered convolutional DNN (based on AlexNet [8]) is the full image.

2.2 Dropout: A Simple Way to Prevent Neural Networks from Overfitting [9]

Dropout is a technique used to overcome the problem of overfitting in deep neural nets with large numbers of parameters. The idea is to train using many "thinned" networks, chosen by randomly removing subsets of units and their connections. The predictions from the thinned networks are approximately averaged at test time by using a single, unthinned, network with reduced weights.

- Existing regularization methods: stopping training as soon as validation error stops improving, L1 and L2 regularization, and weight sharing [10].

2.3 End-to-end people detection in crowded scenes [11]

This paper is focused on jointly creating a set of bounding-box predictions for people in crowded scenes using GoogLeNet and a recurrent LSTM layer as a controller. Since bounding-box predictions are generated jointly, common post-processing steps such as non-maximum suppression are unnecessary. All components of the system are trained end-to-end using back propagation.

The end-to-end people detection method is contrasted with the object detection methods of R-CNN in [7] and OverFeat in [12]. [7] and [12] rely on non-maximum suppression, which does not use access to image information to infer bounding box positions since non-maximum suppression acts only on bounding boxes. Also, in end-to-end people detection, the decoding stage is learned using LSTMs, instead of using specialized methods as in [13] and [14].

Early related work can be found in [15] and [16]. Best performing object detectors at the time were [7], [12], [17], [18] and [19].

Sequence modeling is done using LSTMs as in [20] (used for machine translation) and [21] (used for image captioning). The loss function is similar to the loss function proposed in [22] in that the loss function encourages the model to make predictions in descending order of confidence.

A new training set collected from public webcams, called "Brainwash", is produced. Brainwash consists of 11917 images with 91146 labelled people. 1000 images are allocated for testing and validation, hence training, test and validation sets contain 82906, 4922 and 3318 labels, respectively.

A pre-trained GoogLeNet [23] is used to produce encoded features as input to the LSTM. The GoogLeNet features are further fine-tuned by the training process. Using GoogLeNet, a feature vector of length 1024 is produced for each region over a 15x20 grid of regions that covers the entire 480x640 input image. Each cell in the grid has a receptive field of 139x139, and is trained to produce a set (with fixed cardinality five) of distinct bounding boxes in the center 64x64 region.

At each step, the LSTM for each grid cell, of which there are 300 in total, produces a new bounding box and corresponding confidence that the bounding box contains a person $\boldsymbol{b} = \{\boldsymbol{b}_{pos}, b_c\}$, where $\boldsymbol{b}_{pos} = (b_x, b_y, b_w, b_h) \in \mathbb{R}^4$ and $b_c \in [0, 1]$. The prediction algorithm stops when the confidence drops below a set threshold. The LSTM units have 250 memory states, no bias units, and no output non-linearities. Each LSTM unit adds its output to the image representation, and feeds the result into the next LSTM unit. Comparable results are found by only presenting the image representation as input to the first LSTM unit.

A new loss function that operates on sets of bounding-box predictions is introduced. Denoting bounding boxes generated by the model as $C = \{\tilde{\boldsymbol{b}}_i\}$, and ground truth bounding boxes by $G = \{\boldsymbol{b}_i\}$, the loss function is given by Equation 3.

$$L(G, C, f) = \alpha \sum_{i}^{|G|} l_{pos}\left(\tilde{\boldsymbol{b}}^i_{pos}, \boldsymbol{b}^{f(i)}_{pos}\right) + \sum_{j}^{|C|} l_c\left(\tilde{b}^j_c, y_j\right) \tag{3}$$

In Equation 3, $f(i)$ is an injective function $G \to C$ that assigns one ground truth to each index i up to the number of ground truths, l_{pos} is the L_1 displacement between bounding boxes, and l_c is a cross-entropy loss on a candidate's confidence that a bounding box exists, where $y_j = \mathbb{1}\{f^{-1}(j) \neq \varnothing\}$. α is set to 0.03 from cross-validation.

In creating $f(i)$ in Equation 3 to assign candidate predictions to ground truths, the $G \times C \to \mathbb{R} \times \mathbb{N} \times \mathbb{N}$ function $\Delta\left(\boldsymbol{b}^i, \tilde{\boldsymbol{b}}^j\right) = (o_{ij}, r_i, d_{ij})$ is used to lexicographically order pairs first by o, then r, then d, where o is one if there is sufficient overlap between candidate and ground truth and zero otherwise, r is the prediction's confidence, and d is the L_1 displacement between candidate and ground truth bounding boxes.

With an AP (average precision) of 0.78 and EER (equal error rate) of 0.81, the $f(i)$ produced by minimizing Δ, using the Hungarian algorithm, is found to improve on AP and EER compared with a fixed assignment of $f(i)$, or selecting the first k highest ranked (L_{firstk}). COUNT (Absolute difference between number of predicted and ground truth detections) for $f(i)$ with Hungarian was 0.76 compared with 0.74 for L_{firstk}. As a baseline, Overfeat-GoogLeNet (bounding-box regression on each cell, followed by non-maximum suppression, as in [12]) achieved 0.67, 0.71 and 1.05 AP, EER and COUNT, respectively.

The system is trained with learning rate 0.2, decreased by a factor of 0.8 every 100 000 iterations (with convergence occurring after 500 000 iterations), and momentum 0.5. Gradient clipping is done at 2-norm of 0.1.

Training without finetuning GoogLeNet reduces AP by 0.29.

GoogLeNet activations are scaled down by a factor of 100 before being input to the decoder, since decoder weights are initialized according to a uniform distribution in $[-0.1, 0.1]$, while GoogLeNet activations are in $[-80, 80]$. Regression predictions from GoogLeNet are scaled up by 100 before comparing with ground truth locations (which are in $[-64, 64]$).

Dropout with probability 0.15 is used on the output of each LSTM, removal of which decreases AP by 0.011. Images are jittered by up to 32 pixels in horizontal and vertical directions, and scaled by a factor between 0.9 and 1.1. L_2 regularization of weights in the network was removed entirely. When using the original 2^{-4} L_2 regularization multiplier on GoogLeNet only, the network was unable to train. An L_2 regularization multiplier on GoogLeNet of 10^{-6} reduced AP by 0.03.

It is found that AP (on the validation set) increases from 0.82 to 0.85 when using separate weights connecting each of the LSTM outputs to predicted candidates.

At test time, per-region predictions are merged by adding a new region at each iteration, and destroying any new bounding boxes that overlap previously accepted bounding boxes, under the constraint that any given bounding box can destroy at most one other bounding box. An ordering function $\Delta' : A \times C \rightarrow \mathbb{N} \times \mathbb{R}$ given by $\Delta'(\boldsymbol{b}_i, \tilde{\boldsymbol{b}}_j) = (m_{ij}, d_{ij})$ where m_{ij} denotes intersection of boxes and d_{ij} is L_1 displacement, is minimized using the Hungarian algorithm in order to find a bipartite matching. At each step, any new candidate that is not intersecting in the matching is added to the set of accepted candidates.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," 2016, book in preparation for MIT Press. [Online]. Available: http://www.deeplearningbook.org

[2] J. Goldberger, G. E. Hinton, S. T. Roweis, and R. R. Salakhutdinov, "Neighbourhood components analysis," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. MIT Press, 2005, pp. 513–520. [Online]. Available: http://papers.nips.cc/paper/2566-neighbourhood-components-analysis.pdf

[3] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[4] R. Rothe, M. Guillaumin, and L. J. V. Gool, "Non-maximum suppression for object detection by passing messages between windows," in *Computer Vision - ACCV 2014 - 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part I*, 2014, pp. 290–306. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-16865-4_19

[5] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," *CoRR*, vol. abs/1312.4659, 2013. [Online]. Available: http://arxiv.org/abs/1312.4659

[6] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 2553–2561. [Online]. Available: http://papers.nips.cc/paper/5207-deep-neural-networks-for-object-detection.pdf

[7] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *CoRR*, vol. abs/1311.2524, 2013. [Online]. Available: http://arxiv.org/abs/1311.2524

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*

25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[9] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014. [Online]. Available: http://dl.acm.org/citation.cfm?id=2627435.2670313

[10] S. J. Nowlan and G. E. Hinton, "Simplifying neural networks by soft weight-sharing," *Neural Comput.*, vol. 4, no. 4, pp. 473–493, Jul. 1992. [Online]. Available: http://dx.doi.org/10.1162/neco.1992.4.4.473

[11] R. Stewart and M. Andriluka, "End-to-end people detection in crowded scenes," *CoRR*, vol. abs/1506.04878, 2015. [Online]. Available: http://arxiv.org/abs/1506.04878

[12] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *CoRR*, vol. abs/1312.6229, 2013. [Online]. Available: http://arxiv.org/abs/1312.6229

[13] M. A. Sadeghi and A. Farhadi, "Recognition using visual phrases," 2011.

[14] S. T. nd Mykhaylo Andriluka nd Bernt Schiele, "Detection and tracking of occluded people," *International Journal of Computer Vision (IJCV)*, vol. 110, no. 1, pp. 58–69, 2014.

[15] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2009.167

[16] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, ser. CVPR '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 878–885. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2005.272

[17] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, 2013. [Online]. Available: http://www.huppelen.nl/publications/selectiveSearchDraft.pdf

[18] S. Zhang, R. Benenson, and B. Schiele, "Filtered channel features for pedestrian detection," *CoRR*, vol. abs/1501.05759, 2015. [Online]. Available: http://arxiv.org/abs/1501.05759

[19] C. Szegedy, S. E. Reed, D. Erhan, and D. Anguelov, "Scalable, high-quality object detection," *CoRR*, vol. abs/1412.1441, 2014. [Online]. Available: http://arxiv.org/abs/1412.1441

[20] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *CoRR*, vol. abs/1409.3215, 2014. [Online]. Available: http://arxiv.org/abs/1409.3215

[21] A. Karpathy and F. Li, “Deep visual-semantic alignments for generating image descriptions,” *CoRR*, vol. abs/1412.2306, 2014. [Online]. Available: http://arxiv.org/abs/1412.2306

[22] A. Graves, S. Fernndez, and F. Gomez, “Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks,” in *In Proceedings of the International Conference on Machine Learning, ICML 2006*, 2006, pp. 369–376.

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Computer Vision and Pattern Recognition (CVPR)*, 2015. [Online]. Available: http://arxiv.org/abs/1409.4842