Machine Learning

Survey

I Hinterfragungen

1. Upon reading [@c5], this question arises: Is there any kind of 'specialized neuron or connection or activation function' in a CNN as compared to a NN? Is there any 'hard coding' of some elements to be convolutions, pools, etc.?

Some answers to this are at [@c6, p.37, p.43-45]

2. All the references mentioned in [@c5] can be considered essentials:

1. X. Glorot & Y. Bengio. (2010). Understanding the difficulty of training deep feedforward neural networks. Journal of Machine learning Research.
2. K. Hornik (1991) Approximation Capabilities of Multilayer Feedforward Networks, Neural Networks, 4(2), 251–257
3. A. Maas, A. Hannun, A. Ng (2014). Rectifier Nonlinearities Improve Neural Network Acoustic Models
4. V. Nair, G. Hinton, G (2010). Rectified linear units improve restricted Boltzmann machines (PDF). ICML.
5. Y. Nesterov. A method of solving a convex programming problem with convergence rate O(1/sqr(k)). Soviet Mathematics Doklady, 27:372–376, 1983.
6. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov; Dropout: A Simple Way to Prevent Neural Networks from Overfitting . Journal of Machine Learning Re- search (JMLR).15(Jun):1929−1958, 2014.

3 (DLR)**.** is a very good compass whenever one gets lost.

4. What is the 'bias-variance tradeoff' [@c1, p.421]

II Hinterfragung I

5. What is the answer to the ***covariance shift*** riddle at [@c1, p.41]

6. It is very important to learn the distinction between ***discriminative models*** and ***generative models*** at an early stage. This includes intuition about the difference between joint and conditional probability. In particular why is the former better when the underlying distribution is really complicated? Why is it easier to add prior knowledge to the latter? [@c1, p.48,49,50] . An answer is at 30

7. It is important to (III) to remember (-TR) the difference between nearest and k-nearest neighbor, the latter giving access to a regression average. [@c1, p.52]

8. III-TR the no-free-lunch theorem (search for 'lunch' in tag: ml)

9. III-TR that solving the optimization problem for both lin and nonlin regressions only requires a matrix inversion [@c1, p.61]. III-TR that we get into matrix singularity problems when? [@c1, p.73].

10. ***Underfitting*** means that ***the model is too simple***. ***overfitting*** means that ***the model is too complicated*** with (either of) [@c1, p.74]

- too many parameters or other complications
- insufficient confidence in parameter (e.g failed matrix inverse)
- training error decreases nonetheless? What does this mean?

11. There are 10^{10} (10 billion) useful web pages (that's 10% of all pages). This costs around $1M\$$ per year on EC2. The internet data identity graph has around 10^9 vertices. Crawling twitter would cost $10k\$$ on EC2.

12. What are good references for accelerating building intuition about designing neural network models?

13. " XOR - not linearly separable, Nonlinear separation is trivial, Caveat (Minsky & Papert) Finding the minimum error linear separator is NP hard (this killed Neural Networks in the 70s)." [@c1, p.363]

14. At this point, we stop and take a short look at [@c2]

III Hinterfragung II

15. We survey, as a hook, NNs with deep learning, which is a specific kind of machine learning. [@c2, Chap.5]

16. ***Machine learning is essentially a form of applied statistics with increased emphasis on the use of computers to statistically estimate complicated functions and a decreased emphasis on proving confidence intervals around these functions;***

17. "we therefore present the two central approaches to statistics: frequentist estimators and Bayesian inference." What are the essential differences between the two?

18. "We describe how to combine various algorithm components such as an optimization algorithm, a cost function, a model, and a dataset to build a machine learning algorithm." We need to draw a figure for this!

19. " "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.""

20. "Machine learning tasks are usually described in terms of how the machine learning system should process an example. An example is a collection of features that have been quantitatively measured from some object or event that we want the machine learning system to process. We typically represent an example as a vector $x \in R^n$ where each entry x_i of the vector is another feature. For example, the features of an image are usually the values of the pixels in the image."

21. "Modern object recognition is best accomplished with deep learning (Krizhevsky et al., 2012; Ioffe and Szegedy, 2015)." whose abstract[@c3] says:

" Training Deep Neural Networks is complicated by the fact that the distribution of each layer's inputs changes during training, as the parameters of the previous layers change. This slows down the training by requiring lower learning rates and careful parameter initialization, and makes it notoriously hard to train models with saturating nonlinearities. \emph{We refer to this phenomenon as internal covariate shift, and address the problem by normalizing layer inputs}. Our method draws its strength from making normalization a part of the model architecture and performing the normalization for each training mini-batch. Batch Normalization allows us to use much higher learning rates and be less careful about initialization. It also acts as a regularizer, in some cases eliminating the need for Dropout. Applied to a state-of-the-art image classification model, Batch Normalization achieves the same accuracy with 14 times fewer training steps, and beats the original model by a significant margin. Using an ensemble of batch- normalized networks, we improve upon the best published result on ImageNet classification: reaching 4.9% top-5 validation error (and 4.8% test error), exceeding the accuracy of human raters. "

22. This is a good hook into ***joint probability*** and ***marginalizing*** [@c2, p.115]:

" Classification with missing inputs: Classification becomes more challenging if the computer program is not guaranteed that every measurement in its input vector will always be provided. In order to solve the classification task, the learning algorithm only has to define a single function mapping from a vector input to a categorical output. When some of the inputs may be missing, rather than providing a single classification function, the learning algorithm must learn a set of functions. Each function corresponds to classifying x with a different subset of its inputs missing. This kind of situation arises frequently in medical diagnosis, because many kinds of medical tests are expensive or invasive. One way to efficiently define such a large set of functions is to learn a probability distribution over all of the relevant variables, then solve the classification task by marginalizing out the missing variables. With n input n variables, we can now obtain all 2 different classification functions needed for each possible set of missing inputs, but we only need to learn a single function describing the joint probability distribution. See Goodfellow et al. (2013b) for an example of a deep probabilistic model applied to such a task in this way. ”

23. As a starting answer to 6 in terms of joint and conditional probability, here is what the author says about the difference between super and unsuper learning

“ Roughly speaking, unsupervised learning involves observing several examples of a random vector x, and attempting to implicitly or explicitly learn the probability distribution p(x), or some interesting properties of that distribution, while supervised learning involves observing several examples of a random vector x and an associated value or vector y, and learning to predict y from x, usually by estimating p(y | x). The term supervised learning originates from the view of the target y being provided by an instructor or teacher who shows the machine learning system what to do. In unsupervised learning, there is no instructor or teacher, and the algorithm must learn to make sense of the data without this guide. ”

We have to imagine the unsuper learned distribution of let's say the image of a cat as a probability density, where x (the input) being an actual image of a cat, results in a high $p(x)$, that is $p_{\text{cat}}(x)$. On the other hand the super learned 'thing' is the estimation $p_{\text{cat}}(y|x)$, the probability of label 'cat' given image x.

24. From this comment it seems that we still do not understand the difference between super and unsuper, in particular, in the unsuper example, where do the labels (y) come from?: [@c2, p.120]

“ For example, the chain rule of probability states that for a vector x Rn, the joint distribution can be decomposed as ... This decomposition means that we can solve the ostensibly unsupervised problem of modeling p(x) by splitting it into n supervised learning problems. Alternatively, we can solve the supervised learning problem of learning p(y | x) by using traditional unsupervised learning technologies to learn the joint distribution p(x,y) and inferring... ”

“ Traditionally, people refer to regression, classification and structured output problems as supervised learning. Density estimation in support of other tasks is usually considered unsupervised learning. ”

25. “ What separates machine learning from optimization is that we want the \emph{generalization error}, also called the test error, to be low as well. The generalization error is defined as the expected value of the error on a new input. Here the expectation is taken across different possible inputs, drawn from the distribution of inputs we expect the system to encounter in practice ”

IV Probability and Statistics I

26. This was triggered by the need to understand both conditional probability (question 6) and the likelihood estimator mentioned in [UML] and [DL]

27. ***Pfad***

1. A bit of intuition for statistics with [Wpva34] [...]
2. A great fast survey of statistics that helps easily understand conditional probability and joint distribution with [SP-Ta] [...]
3. An excellent backbone of measure theory with [AMP] [...]
4. After which we can come back to [DL] and [UML] [...]
5. A still unanswered question is the role of 'bayesian' statistics in image classification, for which we found some good sources (tags: ml, mv), hopefully this will clear itself a bit. [...]

V Bayesian A Priori

28. An unexpected path, coming from [AMP], lead us to finally investigate information theory and conceptually crack entropy, a good help was [Information Theory Primer] and the webpage of the author, disambiguating many terms [A Glossary for Biological Information Theory and the Delila System]. The author also lead us to the excellent [Information and Theory] which closes loops and finally hooks up to thermodynamics. Multiple related resources are tagged [IT].

29. A blitz investigation lead us to inverse theory and Tarantola. Here we close many loops, linking physics, probability and differential geometry, and scientific practice. We have wondered about the inverse problem ages ago. We now have the ability to tackle it. [Inverse Problem Theory, and Methods for Model Parameter Estimation] is a true gem, loop closer and connection maker.

30. Looking for an inverse problem point of view on machine learning, we find Poggio, who finally explains in detail some fundamentals (even if outdated). [A Theory of Networks for Approximation and Learning] is gold, it finally also explains what is meant by regularization and relaxation. Also it finally explains 6 in appendix D, which we reproduce here. This is also treated in a more detailed manner in [ITILA, p492, Learning as Inference].

31. Poggio's recent papers also deal with fundamental topics important to us.

VI Kernel, PCA, Covariance

32. A good fast introduction into Kernel methods is the tagged 'Introduction to Kernel Methods'

33. A good way to intuit about variance and covariance are the tagged variance, covariance, intuit

34. A good way to understand PCA and covariance's relation to eigenvectors is to take the Lagrange multiplier route as explained in the tagged 'Probability, covariance, pca, lagrange'.

VII Convolution and (Rotational) (Equi)Invariance

35. The paper [@c4] and its section "Related Work" may be very good pointers.

36. References

1. https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/
2. http://colah.github.io/posts/2014-07-Understanding-Convolutions/
3. http://timdettmers.com/2015/03/26/convolution-deep-learning/
4. https://books.google.de/books?id=bL3SBQAAQBAJ&pg=PA34&lpg=PA34&dq=intuition+machine+vision+convolution&source=bl&ots=5eCP23-MpQ&sig=TinX_zPCBbzqrlsc55hj7-oq-2M&hl=en&sa=X&ved=0ahUKEwjDvcTw9IXQAhWE1BoKHboxDHMQ6AEINDAE#v=onepage&q=intuition%20machine%20vision%20convolution&f=false
5. http://stats.stackexchange.com/questions/168064/why-do-convolutional-neural-networks-not-use-a-support-vector-machi
6. https://en.wikipedia.org/wiki/Convolutional_neural_network
7. https://www.google.de/search?client=safari&rls=en&q=convnet+and+rotated+images&ie=UTF-8&oe=UTF-8&gfe_rd=cr&ei=0LAXWPXaJoXb8AfgjYG4Aw
8. https://www.quora.com/How-is-a-convolutional-neural-network-able-to-learn-invariant-features
9. https://www.quora.com/Why-and-how-are-convolutional-neural-networks-translation-invariant
10. https://www.quora.com/I-was-told-that-pooling-in-a-convolutional-neural-network-leads-to-inva
11. https://www.quora.com/Why-are-normal-neural-networks-not-translation-invariant-but-convolu
12. https://www.quora.com/How-do-you-make-convolutional-neural-networks-invariant-to-scale
13. https://arxiv.org/pdf/1604.06720.pdf
14. https://www.google.de/search?client=safari&rls=en&q=convolutional+neural+network+rotation+invariance&ie=UTF-8&oe=UTF-8&gfe_rd=cr&ei=hbEXWNFbiNvwB7LUhLgK
15. https://arxiv.org/abs/1604.06720
16. http://stackoverflow.com/questions/28201617/rotational-equivariance-in-convolutional-neural-network
17. http://stackoverflow.com/questions/2480650/role-of-bias-in-neural-networks?rq=1
18. http://stackoverflow.com/questions/27280750/trouble-understanding-the-backpropagation-algorithm-in-neural-networ noredirect=1&lq=1

19. http://stackoverflow.com/questions/24545725/deep-belief-networks-vs-convolutional-neural-networks?rq=1
20. http://www.cs.princeton.edu/courses/archive/spr08/cos598B/Readings/Fukushima1980.pdf
21. http://cs231n.stanford.edu/reports2016/107_Report.pdf
22. https://papers.nips.cc/paper/4136-tiled-convolutional-neural-networks.pdf
23. http://ieeexplore.ieee.org/document/7560644/
24. https://arxiv.org/pdf/1411.5908v2.pdf
25. https://www.google.de/search?client=safari&rls=en&q=cnn+pooling+rotation+invariance&ie=UTF-8&oe=UTF-8&gfe_rd=cr&ei=WrQXWOnwJ4WT8QfFgYHgAg
26. https://www.reddit.com/r/MachineLearning/comments/49tg9p/rotation_invariance_in_cnn_no_pooling/
27. https://arxiv.org/pdf/1602.02660v2.pdf
28. http://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Laptev_TI-Pooling_Transformation-Invariant_Pooling_CVPR_2016_paper.pdf
29. https://arxiv.org/pdf/1503.07077v1.pdf
30. http://theorycenter.cs.uchicago.edu/REU/2014/final-papers/sauder.pdf
31. https://arxiv.org/pdf/1203.1513v2.pdf

VIII Help from Morten

37. First tips

1. I think it pays of the know the very basics. All the deep learning stuff is just about constructing one big function (composed in layers/tree like structure), that takes in data (images, video) and computes an output based on its parameters (weights). To find the weights, people use the chain rule to compute its derivatives with respect to the error. This is all there is to backprob. Variants of SGD is the solver of choice in most places. Some use BFGS too.
   - know all the basic layers people use, convolutional, activations (relu) and max-pooling layers, and of course the simple fully connected/dense layers.
   - implement your own full network and optimizer that can be used with mnist or even CIFAR10. Checkout convnetjs for online demos (Karpathy). Reproduce with your own code.
2. Avoid outdated autoencoder work, at least for starters. DL classifiers today are trained from random weights normally, without unsupervised pretraining. Autoencoders are probably fun and interesting but I wouldn't put my focus there in the first go.
3. I find recurrent networks super interesting, because modeling sequences of data has a lot of applications that somehow has a wider reach than looking at images. Checkout LSTMs and other type of recurrent networks and their uses.
4. Take a look at some of the more classic ML stuff like SVMs, random forrests (like xgboost) Those are interesting and widely used algorithms. Up until recently SVM's were thought to more or less outperform neural networks, because people had difficulties getting NNs to work when they got deeper. The new advances with the relu activations, Dropout and pool layers has made the new wave of DL possible, and outperformed the classic methods (at the cost of much more data and training time)
5. Google deepmind is famous for their Reinforcement work with combining Q-learning (RL) with deep NNs. This type of ML is about finding a "policy", a function that maps from some state, and to an optimal choice of action. This is what they used to learn to play Atari games from just looking at the pixels and the game score. This stuff is fun and interesting but I wouldn't focus too much on it, because its applications in the real world are limited at the moment. But still, check out the work of Sudden, the guru of reinforcement learning.
6. Probabilistic programming is something that is pretty cool. Its an area of modern ML that is more well-founded on probability theory and statistics, so maybe it will be to your kind of thing. People use these languages to write programs that can then be "learned" by using statistical sampling

methods like metropolis-hastings and others. I recently saw a talk by josh Tennenbaum where he showed a prob-program generating MNIST digits by "drawing" them. This meant that he estimated the class of the digit by looking at how easy it can be drawn in a continuous motion. http://web.mit.edu/cocosci/josh.html

7. Also, I recommend using Keras (works on top of tensorflow) to do DL and or recurrent networks. Installing tensorflow and cuda etc is a bit of a pain, but there is no easy way at the moment. https://keras.io/
8. ConvnetJS. These demos really capture what DL is all about. From here its just more bells and whistles... :) http://cs.stanford.edu/people/karpathy/convnetjs/

IX Web Sessions

IX.1 Deep Networks

1. Deep Learning
2. deep learning probability distribution - Google Search
3. Amazon.com: deep learning - AI & Machine Learning / Computer Science: Books
4. Python Machine Learning Blueprints: Intuitive data projects you can relate to 1, Alexander Combs, eBook - Amazon.com
5. Machine Learning: Hands-On for Developers and Technical Professionals: Amazon.de: Jason Bell: Fremdsprachige Bücher
6. Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms: Amazon.de: Nikhil Buduma: Fremdsprachige Bücher
7. Artificial Intelligence for Humans, Volume 3: Deep Learning and Neural Networks: Amazon.de: Jeff Heaton: Fremdsprachige Bücher
8. Artificial Intelligence for Humans, Volume 2: Nature-Inspired Algorithms: Amazon.de: Jeff Heaton: Fremdsprachige Bücher
9. architecting neural networks - Google Search
10. Neural Network Architectures
11. https://arxiv.org/pdf/1605.07678v2.pdf
12. https://arxiv.org/pdf/1306.0152.pdf

IX.2 Misc

machine learning - Recognizing image features regardless of their position using Convolutional Neural Networks - Stack Overflow tech-lab.bu.edu/files/resources/articles_cns/MurshedIJCNN99.pdf What is special about rectifier neural units used in NN learning? - Quora Why can the piece-wise linear function (e.g. rectifier) learn nonlinear mapping in artificial neural networks? - Quora Artificial Neural Networks: Does Hebbian Learning rule working on machine learning problems or is it just a theoretical approach? - Quora Where can I get the e-book, 'Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks'? - Quora Neural Smithing What is the best book for learning artificial neural networks? - Quora CS231n Convolutional Neural Networks for Visual Recognition (2) Mathematical Intuition for Performance of Rectified Linear Unit in Deep Neural Networks | Alexandre Dalyac - Academia.edu Neural Networks for Applied Sciences and Engineering: From Fundamentals to ... - Sandhya Samarasinghe - Google Books neural network training as trajectory of point in the space of all possible neural networks - Google Search https://arxiv.org/pdf/1412.6544.pdf https://icmlviz.github.io/assets/papers/24.pdf Generalization of Neural Networks Improve Neural Network Generalization and Avoid Overfitting - MATLAB & Simulink - MathWorks Deutschland Advances in Pattern Recognition Systems Using Neural Network Technologies - Google Books

IX.3 Misc

neural network numerical estimation of gradient - Google Search A Neural Network in 13 lines of Python (Part 2 - Gradient Descent) - i am trask ConvNetJS Trainer Comparison on MNIST 09_Neural_Networks_Learning a random variable is a function - Google Search difference between random variable and probability measure - Google Search distinguishing probability measure, function, distribution - Mathematics Stack Exchange measure theory - What's the difference between a random variable and a measurable function? - Mathematics Stack Exchange Random variable - Wikipedia Probability space - Wikipedia expected value dice roll - Google Search Expected value - Wikipedia

IX.4 Misc

high level questions and answers about neural networks - Google Search Neural Networks: Questions and Answers machine learning - Why are neural networks becoming deeper, but not wider? - Cross Validated https://arxiv.org/pdf/1512.03965v4.pdf What is the recommended minimum training dataset size to train a deep neural network? - Quora Random Ponderings: A Brief Overview of Deep Learning Convolutional Neural Networks (LeNet) — DeepLearning 0.1 documentation Deep Learning Tutorials — DeepLearning 0.1 documentation DeepLearningWorkshopNIPS2007 <Public <TWiki

IX.5 Misc

Maximum likelihood estimation - Wikipedia Likelihood principle - Wikipedia Statistical model - Wikipedia Machine Vision - Wesley E. Snyder, Hairong Qi - Google Books cvn.ecp.fr/personnel/iasonas/course/Lecture_5.pdf www.cs.rpi.edu/~stewart/sltcv/handout-02.pdf szeliski.org/Book/drafts/SzeliskiBook_20100903_draft.pdf citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.2.8991&rep=rep1&type=pdf cntk maximum likelihood estimator - Google Search Maximum Likelihood Estimation | STAT 414 / 415 dice roll maximum likelihood estimator - Google Search https://www2.mrc-lmb.cam.ac.uk/groups/murshudov/content/courses/lmb_crystallography_course_2013/Practicals/LMB-EP-McCoy.pdf maximum likelihood image classification - Google Search Google Image Result for http://image.slidesharecdn.com/machinelearningincomputervision-160609113920-160609131903/95/machine-learning-computer-vision-22-638.jpg?cb=1465478705 Machine learning & computer vision Computer vision: models, learning and inference Chapter 9 Classification Models. - ppt download Google Image Result for http://images.slideplayer.com/15/4559616/slides/slide_27.jpg Redirect Notice Remote Sensing | Free Full-Text | Maximum Likelihood Deconvolution of Beamformed Images with Signal-Dependent Speckle Fluctuations from Gaussian Random Fields: With Application to Ocean Acoustic Waveguide Remote Sensing (OAWRS) Google Image Result for http://www.geo-informatie.nl/courses/grs20306/lectures/08imageprocessingparta/08imagep Maximum likelihood classification Google Image Result for http://nptel.ac.in/courses/105104100/lectureD_28/images/19.gif GPS Surveying Techniques Google Image Result for http://www.nature.com/nmeth/journal/v4/n1/images/nmeth992-F1.jpg Figure - Nature Methods maximum likelihood image classification - Google Search Maximum-Likelihood Image Classification | Digital and Optical Shape Representation and Pattern Recognition | OR88 | SPIE Proceedings | SPIE 11.7 Maximum Likelihood Classifier www.jars1974.net/pdf/12_Chapter11.pdf Random variable - Wikipedia Probability space - Wikipedia Probability density function - Wikipedia Probability distribution - Wikipedia Probability density function - Wikipedia Uniform distribution (continuous) - Wikipedia Probability mass function - Wikipedia Can a Dirac delta function be a probability density function of a random variable? - Mathematics Stack Exchange

IX.6 NN Convolution Intuition

intuition convolution - Google Search An Intuitive Explanation of Convolutional Neural Networks – the data science blog Understanding Convolutions - colah's blog Understanding Convolution in Deep Learning - Tim Dettmers Machine Vision: Theory, Algorithms, Practicalities - E. R. Davies - Google Books svm - Why do Convolutional Neural Networks not use a Support Vector Machine to classify? - Cross Validated Convolutional neural network - Wikipedia convnet and rotated images - Google Search How is a convolutional neural network able to learn invariant features? - Quora Why and how are convolutional neural networks translation-invariant? - Quora I was told that 'pooling' in a convolutional neural network leads to invariance. Is there a rigorous mathematical proof of this? Is this true for any type of pooling or only some types of pooling? - Quora Why are normal neural networks not translation-invariant, but convolutional nets are translation-invariant? - Quora How do you make convolutional neural networks invariant to scale? - Quora https://arxiv.org/pdf/1604.06720.pdf convolutional neural network rotation invariance - Google Search [1604.06720] Learning rotation invariant convolutional filters for texture classification machine learning - rotational equivariance in Convolutional Neural Network? - Stack Overflow artificial intelligence - Role of Bias in Neural Networks - Stack Overflow java - Trouble Understanding the Backpropagation Algorithm in Neural Network - Stack Overflow machine learning - Deep Belief Networks vs Convolutional Neural Networks - Stack Overflow www.cs.princeton.edu/courses/archive/spr08/cos598B/Readings/Fukushima1980.pdf cs231n.stanford.edu/reports2016/107_Report.pdf https://papers.nips.cc/paper/4136-tiled-convolutional-neural-networks.pdf IEEE Xplore Document - Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images https://arxiv.org/pdf/1411.5908v2.pdf cnn pooling rotation invariance - Google Search Rotation invariance in CNN (no pooling)? : MachineLearning https://arxiv.org/pdf/1602.02660v2.pdf www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Laptev_TI-Pooling_Transformation-Invariant_Pooling_CVPR_2016_paper.pdf https://arxiv.org/pdf/1503.07077v1.pdf theorycenter.cs.uchicago.edu/REU/2014/final-papers/sauder.pdf https://arxiv.org/pdf/1203.1513v2.pdf neural network 3d image registration - Google Scholar convolutional neural network image to model registration - Google Search matching images to 3d models - Google Search https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/obj3d_recognition_cvpr2013.pdf

Bibliography

Bengio, Yoshua. 2009. "Learning Deep Architectures for AI." ***Foundations and Trends in Machine Learning*** 2 (1). Now Publishers Inc.: 1–127.

Dieleman, Sander, Jeffrey De Fauw, and Koray Kavukcuoglu. 2016. "Exploiting Cyclic Symmetry in Convolutional Neural Networks." ***CoRR*** abs/1602.02660. http://arxiv.org/abs/1602.02660.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. "Deep Learning." http://www.deeplearningbook.org.

Heaton, Jeff. "The Third Generation of Neural Networks."

"Introduction to Machine Learning, 2013."

Ioffe, Sergey, and Christian Szegedy. 2015. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." ***CoRR*** abs/1502.03167. http://arxiv.org/abs/1502.03167.