CHAPTER

Deep learning of brain images and its application to multiple sclerosis

3

T. Brosch, Y. Yoo, L.Y.W. Tang, R. Tam
The University of British Columbia, Vancouver, BC, Canada

CHAPTER OUTLINE

3.1 INTRODUCTION

Deep learning is a field within machine learning that has been studied since the early 1980s (Fukushima, 1980). However, deep learning methods did not gain in popularity until the late 2000s with the advent of fast general-purpose graphics processors (Raina et al., 2009), layerwise pretraining methods (Hinton et al., 2006; Hinton and Salakhutdinov, 2006), and large datasets (Deng et al., 2009; Krizhevsky et al., 2012). Since then, deep learning methods have become the state-of-the-art in many nonmedical (Krizhevsky et al., 2012; Sainath et al., 2013) and medical (Ciresan et al., 2012; Kamnitsas et al., 2015) applications. There are many different algorithms and models that are commonly referred to as deep learning methods, all of which have two properties in common: (1) the use of multiple layers of nonlinear

Machine Learning and Medical Imaging. http://dx.doi.org/10.1016/B978-0-12-804076-8.00003-7

processing units for extracting features, and (2) the layers are organized to form a hierarchy of low-level to high-level features. Representing data in a feature hierarchy has many advantages for classification and other applications. To give an example of a feature hierarchy, let us consider the domain of face images. The lowest layer of the feature hierarchy is composed of the raw pixel intensities, which are the most basic features of an image. Multiple pixels can be grouped to form general image features like edges and corners, which can be further combined to form face parts such as different variations of noses, eyes, mouths, and ears. Finally, multiple face parts can be combined to form a variety of face images. Learning a feature hierarchy facilitates the parameterization of a large feature space with a small number of values by capturing complex relationships between feature layers. For example, a feature hierarchy consisting of three prototypical shapes for mouths, eyes, ears, and noses is able to represent $3 \times 3 \times 3 \times 3 = 81$ different prototypical faces with only $3 + 3 + 3 + 3 = 12$ features. Without a hierarchical representation of the data, a model would require 81 prototypical face features to span the same face manifold.

In this section, we will introduce the most commonly used deep learning methods for medical image analysis. We start with a description of unsupervised models like restricted Boltzmann machines (RBMs) (Freund and Haussler, 1992; Hinton, 2010), which are the building blocks of deep belief networks (DBNs) (Hinton et al., 2006), a model that can be used for learning a hierarchical set of features from input images without the need for labels. In the second part of the introduction, we will give a brief overview of dense neural networks (DNNs) (Farley and Clark, 1954; Werbos, 1974; Rumelhart et al., 1986) and convolutional neural networks (CNNs) (Fukushima, 1980; LeCun et al., 1989, 1998), which are the most commonly used supervised deep learning methods.

3.1.1 LEARNING FROM UNLABELED INPUT IMAGES

One of the most important applications of deep learning is to learn a feature hierarchy from unlabeled images. The key to learning such a hierarchy is the ability of deep models to be trained layer by layer, where each layer acts as a nonlinear feature extractor. Various methods have been proposed for feature extraction from unlabeled images. In this section, we will first introduce the RBMs (Freund and Haussler, 1992; Hinton, 2010), which are the building blocks of DBNs (Hinton et al., 2006), followed by a short introduction to alternative feature extractors such as stacked denoising autoencoders (SDAEs) (Vincent et al., 2010).

3.1.1.1 From restricted Boltzmann machines to deep belief networks

An RBM is a probabilistic graphical model defined by a bipartite graph as shown in Fig. 3.1. The units of the RBM are divided into two layers, one of visible units $\mathbf{v}$ and the other of hidden units $\mathbf{h}$. There are no direct connections between units within

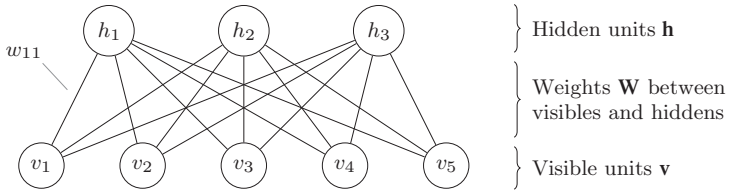

FIG. 3.1

Graphical representation of an RBM with three hidden and five visible units. An RBM models the joint probability of visible and hidden units. Edges between vertices denote conditional dependence between the corresponding random variables.

either layer. An RBM defines the joint probability of visible and hidden units in terms of the energy E:

$$p(\mathbf{v}, \mathbf{h} \mid \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \mathrm{e}^{-E(\mathbf{v},\mathbf{h}|\boldsymbol{\theta})}; \tag{3.1}$$

when the visible and hidden units are binary, the energy is defined as

$$-E(\mathbf{v}, \mathbf{h} \mid \boldsymbol{\theta}) = \sum_{i,j} v_i w_{ij} h_j + \sum_i b_i v_i + \sum_j c_j h_j \tag{3.2}$$

$$= \mathbf{v}^{\mathrm{T}} \mathbf{W} \mathbf{h} + \mathbf{b}^{\mathrm{T}} \mathbf{v} + \mathbf{c}^{\mathrm{T}} \mathbf{h}, \tag{3.3}$$

where $Z(\boldsymbol{\theta})$ is a normalization constant, $\mathbf{W}$ denotes the weight matrix that connects the visible units with the hidden units, $\mathbf{b}$ is a vector containing the visible bias terms, $\mathbf{c}$ is a vector containing the hidden bias terms, and $\boldsymbol{\theta} = \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$ are the trainable parameters of the RBM.

Inference

The hidden units represent patterns of similarity that can be observed in groups of images. Given a set of model parameters $\boldsymbol{\theta}$, the features of an image can be extracted by calculating the expectation of the hidden units. The posterior distribution of the hidden units given the visible units can be calculated by

$$p(h_j = 1 \mid \mathbf{v}, \boldsymbol{\theta}) = \mathrm{sigm}(\mathbf{w}_{\cdot j}^{\mathrm{T}} \mathbf{v} + c_j), \tag{3.4}$$

where $\mathbf{w}_{\cdot j}$ denotes the jth column vector of $\mathbf{W}$ and $\mathrm{sigm}(x)$ is the sigmoid function defined as $\mathrm{sigm}(x) = (1+\exp(-x))^{-1}$, $x \in \mathbb{R}$. An RBM is a generative model, which allows for the reconstruction of an input signal given its features. This is achieved by calculating the expectation of the visible units given the hidden units. The posterior distribution $p(v_i = 1 \mid \mathbf{h}, \boldsymbol{\theta})$ can be calculated by

$$p(v_i = 1 \mid \mathbf{h}, \boldsymbol{\theta}) = \mathrm{sigm}(\mathbf{w}_{i,\cdot}^{\mathrm{T}} \mathbf{h} + b_i), \tag{3.5}$$

where $\mathbf{w}_{i,\cdot}$ denotes the ith row vector of $\mathbf{W}$. Reconstructing the visible units can be used to visualize the learned features. To visualize the features associated with a particular hidden unit, all other hidden units are set to zero and the expectation of the visible units is calculated, which represents the pattern that causes a particular hidden unit to be activated.

Training

RBMs can be trained by maximizing the likelihood or, more commonly, the log-likelihood of the training data, $\mathcal{D} = \{\mathbf{v}_n \mid n \in [1, N]\}$, which is called maximum likelihood estimation (MLE). The gradient of the log-likelihood function with respect to the weights, $\mathbf{W}$, is given by the mean difference of two expectations:

$$\nabla_{\mathbf{W}} \log p(\mathcal{D} \mid \boldsymbol{\theta}) = \frac{1}{N} \sum_{n=1}^{N} \mathbb{E}[\mathbf{v}\mathbf{h}^{\mathrm{T}} \mid \mathbf{v}_n, \boldsymbol{\theta}] - \mathbb{E}[\mathbf{v}\mathbf{h}^{\mathrm{T}} \mid \boldsymbol{\theta}]. \tag{3.6}$$

The first expectation can be estimated using a mean field approximation:

$$\mathbb{E}[\mathbf{v}\mathbf{h}^{\mathrm{T}} \mid \mathbf{v}_n, \boldsymbol{\theta}] \approx \mathbb{E}[\mathbf{v} \mid \mathbf{v}_n, \boldsymbol{\theta}]\mathbb{E}[\mathbf{h}^{\mathrm{T}} \mid \mathbf{v}_n, \boldsymbol{\theta}] \tag{3.7}$$

$$= \mathbf{v}_n \mathbb{E}[\mathbf{h}^{\mathrm{T}} \mid \mathbf{v}_n, \boldsymbol{\theta}]. \tag{3.8}$$

The second expectation is typically estimated using a Monte Carlo approximation:

$$\mathbb{E}[\mathbf{v}\mathbf{h}^{\mathrm{T}} \mid \boldsymbol{\theta}] \approx \frac{1}{S} \sum_{s=1}^{S} \mathbf{v}_s \mathbf{h}_s^{\mathrm{T}}, \tag{3.9}$$

where S is the number of generated samples, and $\mathbf{v}_s$ and $\mathbf{h}_s$ are samples drawn from $p(\mathbf{v} \mid \boldsymbol{\theta})$ and $p(\mathbf{h} \mid \boldsymbol{\theta})$, respectively. Samples from an RBM can be generated efficiently using block Gibbs sampling, in which the visible and hidden units are initialized with random values and alternately sampled given the previous state using

$$h_j = \mathbb{I}(y_j < p(h_j = 1 \mid \mathbf{v}, \boldsymbol{\theta})) \qquad \text{with } y_j \sim \mathrm{U}(0, 1) \tag{3.10}$$

$$v_i = \mathbb{I}(x_i < p(v_i \mid \mathbf{h}, \boldsymbol{\theta})) \qquad \text{with } x_i \sim \mathrm{U}(0, 1), \tag{3.11}$$

where $z \sim \mathrm{U}(0, 1)$ denotes a sample drawn from the uniform distribution in the interval $[0, 1]$ and $\mathbb{I}$ is the indicator function, which is defined as 1 if the argument is true and 0 otherwise. After several iterations, a sample generated by the Gibbs chain is distributed according to $p(\mathbf{v}\mathbf{h} \mid \boldsymbol{\theta})$.

If the Gibbs sampler is initialized at a data point from the training set and only one Monte Carlo sample is used to approximate the second expectation in (3.6), the learning algorithm is called contrastive divergence (CD) (Hinton, 2002). Alternatively, persistent contrastive divergence (PCD) (Tieleman, 2008) uses several separate Gibbs chains to generate data independent samples from the model, which results in a better approximation of the gradient of the log-likelihood than CD. To speed up the training, the dataset is usually divided into small subsets called mini-batches and a gradient step is performed for each mini-batch. To avoid confusion

with a gradient step, the term "iteration" is generally avoided and the term "epoch" is used instead to indicate a sweep through the entire dataset. Additional tricks to monitor and speed up the training of an RBM can be found in Hinton's RBM training guide (Hinton, 2010).

Deep belief networks

A single RBM can be regarded as a nonlinear feature extractor. To learn a hierarchical set of features, multiple RBMs are stacked and trained layer by layer, where the first RBM is trained on the input data and subsequent RBMs are trained on the hidden unit activations computed from the previous RBM. The stacking of RBMs can be repeated to initialize DBNs of any depth.

3.1.1.2 Variants of restricted Boltzmann machines and deep belief networks

Convolutional DBNs

A potential drawback of DBNs is that the learned features are location dependent. Hence, features that can occur at many different locations in an image, such as edges and corners, must be relearned for every possible location, which dramatically increases the number of features required to capture the content of large images. To increase the translational invariance of the learned features, Lee et al. (2009, 2011) introduced the convolutional deep belief network (convDBN). In a convDBN, the units of each layer are organized in a multidimensional array that reflects the arrangement of pixels in the input image. The units of one layer are only connected to the units of a subregion of the previous layer, and share the same weights with all other units of the same layer. This greatly reduces the number of trainable weights, which reduces the risk of overfitting, reduces the memory required to store the model parameters, speeds up the training, and thereby facilitates the application to high-resolution images.

A convDBN consists of alternating convolutional and pooling layers, which are followed by one or more dense layers. Each convolutional layer of the model can be trained in a greedy layerwise fashion by treating it as a convolutional restricted Boltzmann machine (convRBM). The energy of a convRBM is defined as

$$E(\mathbf{v},\mathbf{h}) = -\sum_{i=1}^{N_\mathrm{c}}\sum_{j=1}^{N_\mathrm{k}} \mathbf{h}^{(j)} \bullet (\tilde{\mathbf{w}}^{(ij)} * \mathbf{v}^{(i)}) - \sum_{i=1}^{N_\mathrm{c}} b_i \sum_{x,y=1}^{N_\mathrm{v}} v_{xy}^{(i)} - \sum_{j=1}^{N_\mathrm{k}} c_j \sum_{x,y=1}^{N_\mathrm{h}} h_{xy}^{(j)}. \tag{3.12}$$

The key terms and notation are defined in Table 3.1. At the first layer, the number of channels N_c is one when trained on unimodal images, or equal to the number of input modalities when trained on multimodal images. For subsequent layers, N_c is equal to the number of filters of the previous layer.

Table 3.1 Key Variables and Notation (for Notational Simplicity, We Assume the Input Images to be Square 2D Images)

Symbol	Description
$\mathbf{v}^{(i)}$	A 2D array containing the units of the ith input channel
$\mathbf{h}^{(j)}$	A 2D array containing the units of the jth output channel or feature map
$\mathbf{w}^{(ij)}$	A 2D array containing the weights of filter kernels connecting visible units $\mathbf{v}^{(i)}$ to hidden units $\mathbf{h}^{(j)}$
b_i	Bias terms of the visible units
c_j	Bias terms of the hidden units
N_c	Number of channels of the visible units
N_v	Width and height of the image representing the visible units
N_k	Number of filters and feature maps
N_h	Width and height of a feature map
$\bullet$	Element-wise product followed by summation
$*$	Valid convolution
$\circledast$	Full convolution
$\tilde{\mathbf{w}}^{(ij)}$	Horizontally and vertically flipped version of $\mathbf{w}^{(ij)}$, that is, $\tilde{w}^{(ij)}_{uv} = w^{(ij)}_{N_w-u+1,N_w-v+1}$, where N_w denotes the width and height of a filter kernel

The posterior distributions $p(\mathbf{h} \mid \mathbf{v})$ and $p(\mathbf{v} \mid \mathbf{h})$ can be derived from the energy equation and are given by

$$p(h^{(j)}_{xy} = 1 \mid \mathbf{v}) = \text{sigm}\left(\sum_{i=0}^{N_c-1} (\tilde{\mathbf{w}}^{(ij)} * \mathbf{v}^{(i)})_{xy} + c_j\right), \tag{3.13}$$

$$p(v^{(i)}_{xy} = 1 \mid \mathbf{h}) = \text{sigm}\left(\sum_{j=0}^{N_k-1} (\mathbf{w}^{(ij)} \circledast \mathbf{h}^{(j)})_{xy} + b_i\right). \tag{3.14}$$

To train a convRBM on a set of images $\mathcal{D} = \{\mathbf{v}_n \mid n \in [1, N]\}$, the weights and bias terms can be learned by CD. During each iteration of the algorithm, the gradient of each parameter is estimated and a gradient step with a fixed learning rate is applied. The gradient of the filter weights can be approximated by

$$\Delta\mathbf{w}^{(ij)} \approx \frac{1}{N}(\mathbf{v}^{(i)}_n * \tilde{\mathbf{h}}^{(j)}_n - \mathbf{v}'^{(i)}_n * \tilde{\mathbf{h}}'^{(j)}_n), \tag{3.15}$$

where $\mathbf{h}^{(j)}_n$ and $\mathbf{h}'^{(j)}_n$ are samples drawn from $p(\mathbf{h}^{(j)} \mid \mathbf{v}_n)$ and $p(\mathbf{h}^{(j)} \mid \mathbf{v}'_n)$, and $\mathbf{v}'^{(i)}_n = \mathbb{E}[\mathbf{v}^{(i)} \mid \mathbf{h}_n]$.

Different types of operations (Scherer et al., 2010) have been proposed for the pooling layers, with the common goal of creating a more compact representation of the input data. The most commonly used type of pooling is max-pooling, in which the input to the pooling layer is divided into small blocks and only the maximum

value of each block as passed on to the next layer, which makes the representation of the input invariant to small translations in addition to reducing its dimensionality.

Alternative unit types

To model real-valued inputs like the intensities of some medical images, the binary visible units of an RBM can be replaced with Gaussian visible units, which leads to the following energy function:

$$-E(\mathbf{v}, \mathbf{h} \mid \boldsymbol{\theta}) = \sum_{i,j} \frac{v_i}{\sigma_i} w_{ij} h_j + \sum_i \frac{(v_i - b_i)^2}{2\sigma_i^2} + \sum_j c_j h_j, \tag{3.16}$$

where the mean of the ith visible unit is encoded in the bias term b_i, and its standard deviation is given by σ_i. Although approaches have been proposed to learn the standard deviation (Cho et al., 2011), the training data is often simply standardized to have zero mean and unit variance, which yields the following simplification for the inference of the visible and hidden units:

$$\mathbb{E}[h_j \mid \mathbf{v}, \boldsymbol{\theta}] = \text{sigm}(\mathbf{w}_{\cdot,j}^{\mathrm{T}} \mathbf{v} + c_j), \tag{3.17}$$

$$\mathbb{E}[v_i \mid \mathbf{h}, \boldsymbol{\theta}] = \mathbf{w}_{i,\cdot}^{\mathrm{T}} \mathbf{h} + b_i. \tag{3.18}$$

A binary hidden unit can only encode two states. In order to increase the expressive power of the hidden units, Nair and Hinton (2010) proposed using noisy rectified linear units (NReLUs) as the hidden units, and showed that this can improve the learning performance of RBMs. The signal of an NReLU is the sum of an infinite number of binary units, all of which have the same weights but different bias terms. In the special case where the offsets of their bias terms are set to $-0.5, -1.5, \ldots$, the sum of their probabilities and therefore the expectation of an NReLU is extremely close to having a closed form:

$$\mathbb{E}[h_j \mid \mathbf{v}, \boldsymbol{\theta}] = \sum_{i=1}^{\infty} \text{sigm}(\mathbf{w}_{\cdot,j}^{\mathrm{T}} \mathbf{v} + c_j - i + 0.5) \tag{3.19}$$

$$\approx \log(1 + \exp(\mathbf{w}_{\cdot,j}^{\mathrm{T}} \mathbf{v} + c_j)). \tag{3.20}$$

However, sampling of this type of unit involves the repeated calculation of the sigmoid function, which can be time-consuming. If a sample is not constrained to being an integer, a fast approximation can be calculated with

$$h_j \sim \max(0, \mu_j + \mathcal{N}(0, \text{sigm}(\mu_j))), \tag{3.21}$$

$$\mu_j = \mathbf{w}_{\cdot,j}^{\mathrm{T}} \mathbf{v} + c_j, \tag{3.22}$$

where $\mathcal{N}(0, \sigma^2)$ denotes Gaussian noise.

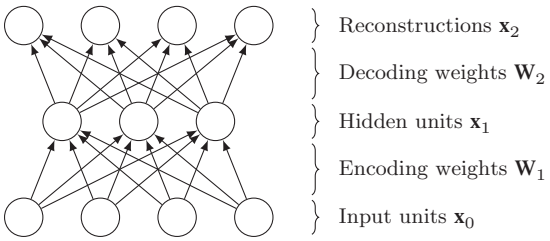

FIG. 3.2

Autoencoder with input units $\mathbf{x}_0$, hidden units $\mathbf{x}_1$, and reconstructions $\mathbf{x}_2$. If the network is trained on corrupted versions of the inputs with the goal of improving the robustness to noise, it is called a denoising autoencoder.

3.1.1.3 Stacked denoising autoencoders

Popular alternatives to DBNs for unsupervised feature learning are stacked autoencoders (SAEs) and SDAEs (Vincent et al., 2010) due to their ability to be trained without the need to generate samples, which speeds up the training compared to RBMs. A minimal autoencoder is a three-layer neural network (see Fig. 3.2) consisting of an input layer $\mathbf{x}_0$, a hidden layer $\mathbf{x}_1$, and an output layer $\mathbf{x}_2$. Similar to RBMs, there are many variants of autoencoders. In the following, we will only consider dense autoencoders with real-valued input units and binary hidden units. Alternative unit types are discussed by Vincent et al. (2010).

The input layer of an autoencoder is a vector containing the intensities of an input image. In the encoding step, features are extracted from the inputs as follows:

$$\mathbf{x}_1 = \text{sigm}(\mathbf{W}_1\mathbf{x}_0 + \mathbf{b}_1), \tag{3.23}$$

where $\mathbf{W}_1$ denotes a matrix containing the encoding weights and $\mathbf{b}_1$ denotes a vector containing the bias terms. In the decoding step, an approximation of the original input signal is reconstructed based on the extracted features:

$$\mathbf{x}_2 = \mathbf{W}_2\mathbf{x}_1 + \mathbf{b}_2, \tag{3.24}$$

where $\mathbf{W}_2$ denotes a matrix containing the decoding weights and $\mathbf{b}_2$ denotes a vector containing the bias terms. An autoencoder is trained by minimizing an error measure (eg, the sum of squared differences or cross-entropy) between the original inputs and their reconstructions. Given a training set $\mathcal{D} = \{\mathbf{x}^{(i)} \mid i \in [1, N]\}$, the optimization problem can be formalized as

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} (\mathbf{x}_2^{(i)} - \mathbf{x}^{(i)})^{\mathrm{T}} (\mathbf{x}_2^{(i)} - \mathbf{x}^{(i)}), \tag{3.25}$$

where $\mathbf{x}_0^{(i)} = \mathbf{x}^{(i)}$ and $\boldsymbol{\theta} = \{\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2\}$ are the parameters of the autoencoder. The optimization problem can be solved using stochastic gradient descent (SGD)

(Rumelhart et al., 1986) (see Section 3.1.2.1). If the hidden layer contains fewer units than the input layer, the autoencoder learns a lower-dimensional representation of the input data, which allows the model to be used for dimensionality reduction.

The learning of the features can be improved by altering the input signal with random perturbations such as adding Gaussian noise or randomly setting a fraction of the input units to zero. This forces the model to learn features that are robust to noise and capture structures that are useful for reconstructing the original signal. An autoencoder trained on the corrupted versions of the input images is called a denoising autoencoder. Similar to DBNs, a stack of autoencoders can learn a hierarchical set of features, where subsequent autoencoders are trained on the extracted features of the previous autoencoder.

3.1.2 LEARNING FROM LABELED INPUT IMAGES

Features extracted by unsupervised feature learning methods are often fed into a separate supervised learning model, such as a random forest (Breiman, 2001a) or support vector machine (Cortes and Vapnik, 1995), to perform classification or prediction. Alternatively, classification and prediction can be performed with a single model that takes the raw input data and produces the desired output, such as class probabilities. This type of learning is called end-to-end learning and has shown great potential for medical image analysis (Ciresan et al., 2012). The most popular models for end-to-end learning are neural networks due to their ability to learn a hierarchical set of features from raw input data. The supervised learning framework allows for the learning of features that are tuned for a given combination of input modalities and classification tasks, but is more prone to overfitting than unsupervised feature learning, especially when the amount of labeled data is limited. In this section, we will start with an introduction to DNN, followed by a concise overview of CNNs.

3.1.2.1 Dense neural networks

A DNN is a deterministic function that maps input data to the desired outputs through the successive application of multiple nonlinear mappings of the following form:

$$\mathbf{z}_l = \mathbf{W}_l \mathbf{x}_{l-1} + \mathbf{b}_l, \tag{3.26}$$

$$\mathbf{x}_l = f_l(\mathbf{z}_l), \tag{3.27}$$

where l indexes a unit layer, $\mathbf{x}_0$ denotes a vector containing the input of the neural network, $\mathbf{x}_L$ denotes a vector containing the output, L is the number of computational layers, f_l are transfer functions, $\mathbf{W}_l$ are weight matrices, and $\mathbf{b}_l$ are bias terms. Popular choices for the transfer function are the sigmoid function $f(x) = \text{sigm}(x)$ and the rectified linear function $f(x) = \max(0, x)$. The same transfer function is typically used for all layers except for the output layer. The choice of the output transfer function depends on the learning task. For classification, a 1-of-n encoding

of the output class is usually used in combination with the softmax transfer function defined as

$$\text{softmax}(\mathbf{a})_i = \frac{\exp(a_i)}{\sum_{j=1}^{n} \exp(a_j)}, \tag{3.28}$$

where $\mathbf{a}$ denotes an n-dimensional output vector.

Given a training set $\mathcal{D} = \{(\mathbf{x}_0^{(i)}, \mathbf{y}^{(i)}) \mid i \in [1, N]\}$, a neural network is trained by minimizing the error between the predicted outputs $\mathbf{x}_L^{(i)}$ and the given labels $\mathbf{y}^{(i)}$:

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \sum_{i=1}^{N} E\left(\mathbf{x}_L^{(i)}, \mathbf{y}^{(i)}\right), \tag{3.29}$$

where $\boldsymbol{\theta}$ denotes the trainable parameters of the neural network. Typical choices for the error function are the sum of squared differences (SSD) and cross-entropy. The minimization problem can be solved using SGD (Rumelhart et al., 1986; Polyak and Juditsky, 1992), which requires the calculation of the gradient of the error function with respect to the model parameters. The gradient can be calculated by backpropagation (Werbos, 1974) as follows:

$$\boldsymbol{\delta}_L = \nabla_{\mathbf{x}_L} E \cdot f_L'(\mathbf{z}_L), \tag{3.30}$$

$$\boldsymbol{\delta}_l = (\mathbf{W}_{l+1}^{\mathrm{T}} \boldsymbol{\delta}_{l+1}) \cdot f_l'(\mathbf{z}_l) \quad \text{for } l < \mathrm{L}, \tag{3.31}$$

$$\nabla_{\mathbf{W}_l} E = \boldsymbol{\delta}_l \mathbf{x}_{l-1}^{\mathrm{T}}, \tag{3.32}$$

$$\nabla_{\mathbf{b}_l} E = \boldsymbol{\delta}_l, \tag{3.33}$$

where $\nabla_{\mathbf{x}_L} E$ denotes the gradient of the error function with respect to the predicted output and $\cdot$ denotes element-wise multiplication.

3.1.2.2 Convolutional neural networks

The structure of CNNs is inspired by the complex arrangement of simple and complex cells found in the visual cortex (Hubel and Wiesel, 1962, 1968). Simple cells are only connected to a small subregion of the previous layer and need to be tiled to cover the entire visual field. In a CNN (see Fig. 3.3), simple cells are represented by convolutional layers, which exhibit a similar mechanism of local connectivity and weight sharing. Complex cells combine the activation of simple cells to add robustness to small translations. These cells are represented in the form of pooling layers similar to the pooling layers found in convDBNs. After several alternating convolutional and pooling layers, the activations of the last convolutional layer are fed into one or more dense layers to carry out the final classification.

For multimodal 3D volumes, the neurons of convolutional and pooling layers are arranged in a 4D array, where the first three dimensions correspond to the dimensions of the input volume, and the fourth dimension indexes the input modality or channel. The activations of the output of a convolutional layer are calculated by

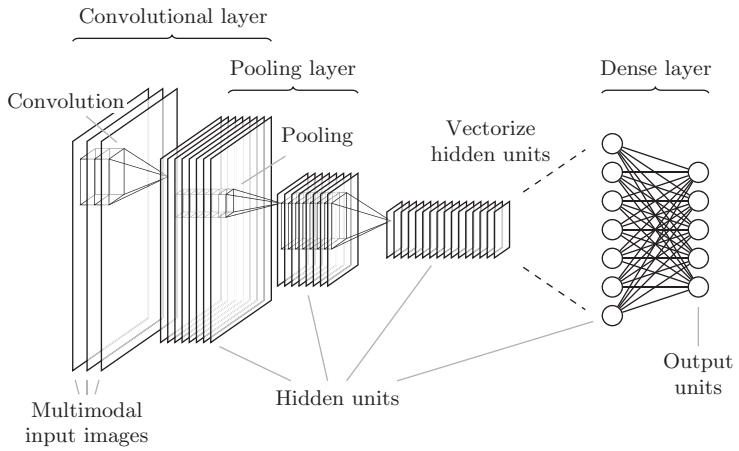

FIG. 3.3

Convolutional neural network with two convolutional layers, one pooling layer and one dense layer. The activations of the last layer are the output of the network.

$$x_j^{(l)} = f\left(\sum_{i=1}^{C} \tilde{w}_{ij}^{(l)} * x_i^{(l-1)} + b_j^{(l)}\right), \tag{3.34}$$

where l is the index of a convolutional layer, $x_j^{(l)}$ denotes the jth channel of the output volume, $w_{ij}^{(l)}$ is a 3D filter kernel connecting the ith channel of the input volume to the jth channel of the output volume, $b_j^{(l)}$ denotes the bias term of the jth output channel, and $\tilde{w}$ denotes a flipped version of w, that is, $\tilde{w}(a) = w(-a)$. CNNs can be trained using SGD, where the gradient can be derived analogously to DNNs and calculated using backpropagation (LeCun et al., 1989, 1998).

A major challenge for gradient-based optimization methods is the choice of an appropriate learning rate. Classic SGD (LeCun et al., 1998) uses a fixed or decaying learning rate, which is the same for all parameters of the model. However, the partial derivatives of parameters of different layers can vary substantially in magnitude, which can require different learning rates. In recent years, there has been an increasing interest in developing methods for automatically choosing independent learning rates. Most methods (eg, AdaGrad by Duchi et al. (2011); AdaDelta by Zeiler (2012); RMSprop by Dauphin et al. (2015); and Adam by Kingma and Ba (2014)) collect different statistics of the partial derivatives over multiple iterations and use this information to set an adaptive learning rate for each parameter. This is especially important for the training of deep networks, where the optimal learning rates often differ greatly for each layer.

3.2 OVERVIEW OF DEEP LEARNING IN NEUROIMAGING

Deep learning methods have been applied with great success to a number of image understanding tasks, such as the detection, segmentation, and classification of objects and regions in images. They have demonstrated impressive improvements over the traditional machine learning techniques and have approached human-level performance in visual and sound recognition applications (LeCun et al., 2015). Due to these successes, deep learning has attracted the attention of neuroimaging researchers to investigate the potential of deep learning for analyzing medical images such as computed tomography (CT), magnetic resonance imaging (MRI), positron emission tomography (PET), and so on.

More specifically, there are two main reasons why deep learning methods are seen as highly promising for neuroimage analysis. First, neuroimaging data is generally high-dimensional. For example, MR images are typically 3D volumes and contain several million voxels each. In addition, due to the continuing development of advanced MRI techniques such as diffusion tensor imaging (DTI) (Le Bihan et al., 2001), functional MRI (Song et al., 2006), susceptibility weighted imaging (SWI) (Haacke et al., 2004), myelin water imaging (MWI) (Alonso-Ortiz et al., 2015), etc., current MRI datasets often contain several modalities that provide complementary information, which should ideally be analyzed together, at the cost of even greater dimensionality. The capability of deep learning to automatically capture discriminative and abstract features through a hierarchical manner may prove particularly useful for reducing the high dimensionality of neuroimaging data to extract key patterns that are representative of important variations. Second, neuroimaging data often lacks labeled data, because labels typically require expert annotations, which can be time-consuming and expensive to obtain. Compared to the huge datasets of natural images often used in the machine learning community that often contain hundreds of thousands of labeled training samples, a neuroimaging dataset with several hundred labeled training samples is already considered large. The limited amount of labeled training data is a common cause of overfitting, making methodological development for neuroimaging application challenging. Many deep learning models can be trained in an unsupervised fashion to learn a representative feature set, which can then be used to initialize a supervised model that is trained with a smaller set of labeled images. By starting with a more generalized set of features, a supervised model can potentially be more robust to overfitting.

The current main applications of deep learning to neuroimaging are segmentation and classification, although there has also been some work on image registration. In neuroimaging, segmentation is typically used to extract desired structures or regions of the central nervous system (CNS) such as white matter (WM), gray matter (GM), cerebrospinal fluid (CSF), spinal cord, corpus callosum, etc., of which volume or shape can be subsequently used to perform diagnosis, monitor disease progress, or study its pathology. Neuroimage classification can be used to perform automatic computer-aided diagnosis, clinical prediction and early disease detection. Traditional machine learning techniques for neuroimage classification generally require

hand-crafted features with domain knowledge or assumptions about disease pathology, which may be subject to bias. Several recent articles have shown that deep learning can automatically learn discriminative features for classification with little prior knowledge. The remainder of this section will describe the recent applications of deep learning for medical image registration, segmentation, and classification that are focused on neuroimaging.

3.2.1 DEFORMABLE IMAGE REGISTRATION USING DEEP-LEARNED FEATURES

Deformable image registration is an important process in many neurological studies for determining the anatomical correspondences which can be used, for example, for atlas-based segmentation of brain structures. The principle behind deformable image registration is to determine the optimal transformation that maximizes the feature similarities between two images, which often relies on user-selected features such as Gabor filters. Wu et al. (2013) proposed using an unsupervised two-layer stacked convolutional independent subspace analysis (ISA), which is an extension of independent component analysis (ICA), to directly learn the basis image filters that represent the training dataset. During image registration, the coefficients of these learned basis filters are used as morphological signatures to detect the spatial correspondences. When incorporated into existing registration methods, the data-adaptive features learned from the unsupervised deep learning framework gave rise to improved results over the hand-crafted features.

3.2.2 SEGMENTATION OF NEUROIMAGING DATA USING DEEP LEARNING

In machine learning, the most common approach to the image segmentation problem consists of two stages: hand-designed feature detectors are used to build feature vectors for each input in the first step, sometimes with prior knowledge such as spatial regularization or contour smoothness, and then the extracted features and target labels are used to train a supervised classifier to perform segmentation. This generally requires much labeled training data and good domain knowledge to design or select features, such as Gabor filters (Jain and Farrokhnia, 1990), Haar wavelet (Mallat, 1989), and SIFT (Lowe, 1999). However, such features are designed based on users' prior knowledge. With the hypothesis that the most effective features could be learned directly from training data, a number of researchers have recently adopted the deep learning framework as discussed below.

3.2.2.1 Hippocampus segmentation

Several methods have been developed for automatic hippocampus segmentation in MR images, as measurement of the hippocampus is useful for studying many neurological diseases including Alzheimer's disease (AD). However, segmentation

accuracy is often limited due to the small size of the hippocampus and the complexity of surrounding structures. As similarly done in Wu et al. (2013), Kim et al. (2013) proposed integrating an unsupervised two-layer stacked convolutional ISA into a multi-atlas-based segmentation framework. The authors compared the traditional hand-crafted image features with the hierarchical feature representations learned from 7.0T MR images. They showed that the deep-learned feature representation improved the segmentation accuracy by about 3–4% on overlap metrics over the methods using hand-crafted features in the same segmentation framework. Guo et al. (2014) investigated using a two-layer SAE to learn the features for segmenting the hippocampus from infant T1- and T2-weighted (T1w and T2w) brain MR images. The deep-learned features were used to measure inter-patch similarity for sparse patch matching in a multi-atlas-based segmentation framework, and demonstrated an improvement of 4–8% in Dice similarity over features based on intensity, Haar wavelet, histogram of oriented gradients (HOG) (Dalal and Triggs, 2005) and local gradient patterns (Ojala et al., 2002).

3.2.2.2 Infant brain image segmentation

In studying early brain development in health and disease, segmenting infant brain images is a more challenging task than in the adult brain because infant WM and GM exhibit similar intensity levels in both T1w and T2w MR images. Zhang et al. (2015) proposed employing a deep CNN to perform segmentation of brain tissues on T1w, T2w and fractional anisotropy (FA) MR images. They trained a three-layer CNN using approximately 10,000 local patches extracted from all voxels in a training set of 10 brain images. The cross-entropy loss functional between the predicted and ground truth labels, and a three-way softmax layer were used to generate a posterior distribution over the three class labels.

3.2.2.3 Brain tumor segmentation

Automatic segmentation of brain tumors is a challenging problem because the tumors can appear randomly in the brain and have any kind of shape, size, and contrast. Havaei et al. (2015) proposed a fully automatic brain tumor segmentation method based on deep CNNs. The proposed method was designed to detect both low- and high-grade glioblastomas seen in MRI scans. The proposed deep CNNs learn contextual features at different scales and are fully convolutional, in contrast to traditional CNNs whose final layers are typically fully connected and therefore much more computationally demanding. In brain tumor segmentation, the training samples are highly unbalanced in that the healthy voxels comprise a large percentage of the total brain voxels. The authors tackled this problem by introducing a sequential training procedure in which the model was first trained with unbalanced training data using randomly sampled voxels, and then the top layers of the model were trained with balanced samples that contained the same number of tumor and healthy voxels.

3.2.3 CLASSIFICATION OF NEUROIMAGING DATA USING DEEP LEARNING

Classification of human neuroimaging data has been typically used to demonstrate the performance of a proposed hand-crafted feature (eg, a set of voxel intensities or size of particular regions of interest) or a feature selection method, both of which often require a thorough understanding of the disease by the user. Deep learning algorithms have recently attracted considerable attention from neuroimaging researchers due to the promise of automatic feature discovery for the tasks of computer-aided disease diagnosis and prognosis. Several recent studies have shown that deep learning methods can improve neuroimaging data classification by learning physiologically important image feature representations and discovering multimodal latent patterns in a data-driven way, as discussed below.

3.2.3.1 Schizophrenia diagnosis

Plis et al. (2014) adapted RBMs for performing schizophrenia diagnosis using the structural brain MR images from 198 schizophrenia patients and 191 matched controls. The MR images were aligned to brain templates and their gray matter was segmented, which resulted in 60,465 voxels per image. The gray matter voxels were vectorized and used to train a three-layer DBN. The first two layers had 50 hidden units in each layer and 100 hidden units were used for the third layer. They pretrained each layer via an unsupervised RBM and discriminatively fine-tuned the network by adding a softmax layer on top of the model using backpropagation. Using activations of the topmost hidden layers in the fine-tuned model with a 10-fold cross-validation, they trained and tested supervised classifiers, such as support vector machine (SVM) with the radial basis function kernel, logistic regression and *k*-nearest neighbors. The effect of the model depth on classification accuracy was investigated. The accuracy remained almost the same from depth 1 to depth 2 (66% and 62%, respectively, using the SVM classifier), but significantly improved for depth 3 (90% using the SVM classifier). Even though the model did not improve from depth 1 to depth 2, the model continued to learn useful transformations of the training data. This result strengthens the hypothesis that unsupervised pretraining can potentially lead to progressively more discriminative features at higher layers of data representation.

3.2.3.2 Huntington disease diagnosis

Plis et al. (2014) used the same three-layer DBN model described in Section 3.2.3.1 to investigate its potential for diagnosing Huntington disease. The dataset in the study was unbalanced, and consisted of 1.5 T and 3.0 T T1w MR images collected from 2641 patients and 859 healthy controls. Similarly to the previous study, the segmented gray matter voxels were utilized to train the deep learning network. The learned network performed binary classification as well as distinguished the patients by disease severity, which enabled spectral decomposition of the images using regression on the learned features.

3.2.3.3 Task identification using functional MRI dataset

ICA is the most widely used method for identifying the most salient signals in functional MRI data. Hjelm et al. (2014) proposed using a Gaussian-Bernoulli RBM model to isolate linear factors in functional brain imaging data by fitting a probability distribution model to the data, in order to identify functional networks. A voxel-by-time data matrix was utilized to train the Gaussian-Bernoulli RBM model. Various aspects of analyzing functional networks and temporal activations were considered for comparing between RBMs and ICA, which led the authors to conclude that an RBM can be used to perform functional network identification with accuracy that is equal to or greater than that of ICA.

3.2.3.4 Early diagnosis of Alzheimer's disease

Making an accurate early diagnosis of AD is particularly important because awareness of the severity and progression risks may enable early treatment. Suk et al. (2015) proposed a three-layer SAE model for classifying between AD and mild cognitive impairment (MCI), a prodromal stage of AD. Gray matter tissue volumes from MRI, mean signal intensities from PET, and biological measures from CSF samples were used as features for training modality-specific deep learning networks. The learned feature set was then reduced using group lasso (Yuan and Lin, 2006), which regularizes a linear regression model with L_1- and $L_{2,1}$-norm. Finally, a multikernel SVM (Gönen and Alpaydın, 2011) designed to learn the complementary information from multimodal data was trained with the learned MRI, PET, CSF features and labels to perform the following classification tasks: (1) AD vs. healthy normal control (NC); (2) MCI vs. NC; (3) AD vs. MCI;, and (4) MCI converter (MCI-C) vs. MCI nonconverter (MCI-NC). The proposed method demonstrated accuracy rates of 98.8%, 90.7%, 83.7%, and 83.3% for AD/NC, MCI/NC, AD/MCI, and MCI-C/MCI-NC classification, respectively, on the scans of 51 AD patients, 99 MCI patients (43 MCI-Cs and 56 MCI-NCs), and 52 NC subjects from the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset. A limitation of this study was that since structural and functional changes involved in AD can occur in multiple brain regions that do not necessarily correspond to user-defined regions of interest (ROIs), the features extracted from such user-defined ROIs may not be able to reflect small or subtle but potentially important pathological changes. Motivated by this, Suk et al. (2014) in a follow-up study proposed using a patch-based multimodal DBM framework (Salakhutdinov and Hinton, 2012) to learn a joint spatial feature representation from the paired 3D patches of MRI and PET images. In contrast to RBMs, the approximate inference procedure of DBMs is performed using two-way dependencies, that is, bottom-up and top-down, which allows DBMs to use higher-level information to learn intermediate-level features, thus creating potentially more accurate representations of the data (Salakhutdinov and Hinton, 2012). Unlike the previous work (Suk et al., 2015), CSF measures were not used in the study. The learned MRI-PET feature representation was then used to train an image-level supervised classifier based on weighted ensemble SVMs. The authors reported accuracy rates of 95.35%, 85.67%, and 74.58% for AD vs. NC, MCI vs. NC,

and MCI-C vs. MCI-NC classification, respectively. Liu et al. (2014a) investigated a deep learning architecture consisting of an SAE with a softmax output layer for performing four-class classification simultaneously for the following labels: AD, NC, MCI-C, and MCI-NC. The mean MR and PET intensity values extracted from 83 brain ROIs were used as features. A multimodal layer was included in the deep learning framework in a denoising fashion, which hid one modality in some samples during training for regularization. The softmax layer had four units for representing the four AD categories. A mean accuracy of 53.79% for four-class classification was reported on 331 scans of the ADNI dataset.

3.2.3.5 High-level 3D PET image feature learning

Liu et al. (2014b) developed a framework based on SAEs to extract high-level ROI features from 3D PET images. The learned feature parameters were used as the encodings for content-based retrieval with the *k*-nearest-neighbor algorithm. The method was evaluated on mean average precision (MAP) using the leave-one-out paradigm with 331 3D PET images from the ADNI cohort. It was shown that the high-level PET ROI features extracted by deep learning can achieve an overall MAP of 56.13%, which outperformed the most widely used state-of-the-art data representation methods, such as Isomap (Tenenbaum et al., 2000) and elastic net (Shen et al., 2011).

3.3 FOCUS ON DEEP LEARNING IN MULTIPLE SCLEROSIS

3.3.1 MULTIPLE SCLEROSIS AND THE ROLE OF IMAGING

Multiple sclerosis (MS) is a chronic, degenerative disease of the brain and spinal cord. The clinical presentation of MS is very heterogeneous, and the range and severity of symptoms can vary greatly between patients. The clinical course of MS is highly unpredictable, but most patients are initially diagnosed as having relapsing remitting MS (RRMS), which is characterized by inflammatory attacks separated by variable periods of remission and recovery. The majority of RRMS patients will eventually transition into the secondary progressive MS (SPMS) phase, in which there is an unremitting and progressive accumulation of disability. There is currently no cure for MS. Existing therapies that focus on symptomatic management and prevention of further damage have variable degrees of effectiveness, although several recent breakthroughs are promising. MS pathology originates at the cellular level and many aspects are not well understood, but there are characteristic (but not specific) signs of tissue damage, the most recognizable of which are white matter lesions (WMLs) and brain atrophy, or shrinkage due to degeneration. These signs can be observed on MRI, which has become a vital tool to noninvasively monitor MS patients in the clinic and to advance the understanding of MS pathology. WML counts and volume and brain volume have become established imaging biomarkers for MS clinical trials, and there is promise for their use in routine clinical practice,

but they generally only correlate modestly with clinical disability scores. The weak link between the image-based measures of MS pathology and disability scores is known as the "clinico-radiological paradox" of MS (Barkhof, 2002), and results in the low utility of current imaging biomarkers for the purposes of personalized medicine. There are a number of key reasons why the current imaging biomarkers do not have stronger quantitative relationships with clinical scores:

- Due to the wide range of symptoms, MS disability is difficult to score comprehensively in routine clinical practice. For example, the Kurtzke (1983) expanded disability status scale (EDSS) is the most commonly used clinical score, but does not account for cognitive impairment, which is a significant contributor to disability in the majority of MS patients (Chiaravalloti and DeLuca, 2008).
- Through neuroplasticity, the brain and spinal cord can adapt to damage in order to maintain functionality (Tomassini et al., 2012). As a result, clinically silent or subtle pathology is often present.
- Conventional MRI does not capture all aspects of MS pathology. For example, myelin is a nerve insulator that is critical for proper signal conduction and demyelination is a key pathological feature of MS in which white matter that appears normal on conventional MRI may actually have reduced myelin (Laule et al., 2004).
- The current established imaging biomarkers largely capture volumetric changes, which are important and relatively easy to compute, but do not reflect potentially important structural variations, such as shape changes in the brain and spatial dispersion of the lesions.
- Traditional statistical approaches like simple prediction models such as logistic regression are common tools for analysis. The general assumption behind these simple models is that the data is generated by a known stochastic data model, with the goodness-of-fit evaluated by residual analysis, but this works well only with a very low number of variables (Breiman, 2001b). Consequently, these traditional statistical approaches, which place a strong emphasis on interpretability, often come with the sacrifice of accuracy.

In this section, we summarize the recent work on deep learning methods for discovering image features relevant to MS, which hold the promise of overcoming the last two limitations listed above. In particular, we focus on two applications: segmentation of WMLs and modeling of disease variability. The overall goal is to investigate the potential of deep learning to automatically capture image features with as little user bias as possible (such as in the choice of model distribution or feature representation), in a way that is complementary to the traditional approach of proposing and validating imaging biomarkers based on biological hypotheses. By performing unsupervised learning on the very high-dimensional space of 3D brain MRIs, deep learning can potentially be a powerful method for generating hypotheses that can then be investigated with more traditional means to facilitate interpretation.

3.3.2 WHITE MATTER LESION SEGMENTATION

Focal lesions in the brain and spinal cord are one of the hallmarks of MS pathology, and are primarily visible in the white matter on structural MRIs. These lesions are observable as hyperintensities on T2w, proton density-weighted (PDw), or fluid-attenuated inversion recovery (FLAIR) scans, and as hypointensities, or "black holes" (García-Lorenzo et al., 2013) on T1w scans. Imaging biomarkers based on the identification of lesions, such as lesion count and lesion volume, have established their importance for assessing disease progression and treatment effect. However, lesions vary greatly in size, shape, intensity and location, which makes their automatic and accurate segmentation challenging. Many automatic methods have been proposed for the segmentation of MS lesions over the last two decades, most of which are described in recent surveys by García-Lorenzo et al. (2013) and Lladó et al. (2012). Lesion segmentation methods can be broadly classified into unsupervised and supervised methods. Unsupervised methods do not require a labeled dataset for training, but most will rather model healthy tissue (eg, via intensity clustering) and identify lesions as an outlier class. Supervised approaches typically start with a set of features, sometimes very large, which are usually defined or selected by the user. A training step with labeled data is used to determine which subset of features produce the most accurate segmentations, and those features are then used to identify lesion voxels in new images.

Given the ability to automatically learn useful features demonstrated by deep learning in various computer vision applications, it is not surprising that a number of MS lesion segmentation methods based on deep learning have been proposed. The potential advantage of deep learning is that the feature set would no longer need to be predetermined by the user, but rather learned directly from the training images. This is a useful property because it is difficult for a person to characterize the features that separate lesion voxels from those of healthy tissue. From the perspective of the deep learning researcher, the high-dimensionality of the input images, the difficulty of obtaining reliable ground truth (thereby making unsupervised learning even more important), and the high accuracy required for clinical practicality all make WML segmentation a worthy test application. In this section, we summarize several deep learning methods that have been recently proposed for MS lesion segmentation. We refrain from listing the values of the performance measures (eg, Dice coefficient) given in the cited papers, because these numbers cannot be compared across datasets, and a publicly available dataset with an adequate sample size for deep learning is sadly not yet available. Therefore we instead focus our summary on the overall observations made by the respective authors. Unless otherwise noted, the experiments described below used MRIs with a voxel size of approximately $1\,\text{mm} \times 1\,\text{mm} \times 3\,\text{mm}$.

3.3.2.1 Patch-based segmentation methods

Yoo et al. (2014) were the first to propose an automated feature learning approach for MS lesion segmentation. In this method, uniformly spaced and nonoverlapping 3D

patches at two scales ($9\times9\times3$ and $15\times15\times5$) were extracted from co-registered T2w and PDw images. Unsupervised feature learning was applied to the T2w and PDw images separately, using a basic architecture of a single RBM for the smaller patches and a two-layer DBN for the larger patches for each image type. Sigmoid hidden units and Gaussian visible units were used, with contrastive divergence for training. The learned features for each image pair were then concatenated into an approximately 5000-element vector that was then fed into a random forest, which was trained with voxel labels for supervised learning. The authors varied the number of image pairs used for unsupervised training from 100 to 1400, while keeping the number of labeled pairs constant at 100, with the hypothesis that feature learning from unlabeled images can improve performance. While the results were not definitive, there was a trend toward improved segmentation accuracy. Overall, this study established that automatic feature discovery with deep learning is a viable alternative to user preselection of features for WML segmentation.

In 2015, the International Symposium on Biomedical Imaging (ISBI) conference held a grand challenge on longitudinal MS lesion segmentation, and released a dataset composed of training data with longitudinal images (with manual delineations of WMLs produced by two raters) from five patients, and two test datasets consisting of longitudinal images from ten and five patients. Each longitudinal dataset included T1w, T2w, PDw, and FLAIR MRIs with three to five time points acquired on a 3T MR scanner. The T1w images had approximately a 1 mm cubic voxel resolution. The winner of the challenge was a team (Vaidya et al., 2015) that used 3D CNNs to automatically learn features from 3D four-channel patches of size $19 \times 19 \times 19$. The basic network architecture was composed of two convolutional layers with 60 ($4 \times 4 \times 4$) and 60 ($3 \times 3 \times 3$) filters using the softplus activation function and average pooling ($2\times2\times2$), followed by a fully connected layer and finally a softmax layer for voxel-wise classification. Gradient descent using log-likelihood as the cost function and momentum was used for training. Since MS lesions typically comprise a very small percentage ($<1\%$) of the voxels in an MRI volume, the patches were selected by dividing each image volume into subregions, and only using patches from those subregions that have greater than a given percentage of lesion voxels. This method obtained segmentation performance comparable to the variability between the two manual raters. It is notable that a deep learning method was able to win the challenge using such a small number of training images. Another method using CNNs (Ghafoorian and Platel, 2015) was also presented at the ISBI challenge. This method used 2D (32×32), four-channel patches, obtained in a sliding window manner and sampled to maintain the proportion between the positive and negative voxels. The network architecture was a CNN consisting of four convolutional layers with 15 (13×13), 25 (9×9), 60 (7×7), and 130 (3×3) filters, with no pooling, and a final logistic regression layer. SGD with a fixed learning rate was used for training. Although the highly competitive performance of the first method is a positive indication for the use of deep learning for WML segmentation, the difference in performance between the two similar CNN-based methods highlights the lack of standardization for the design and application of such models in neuroimaging.

3.3.2.2 Convolutional encoder network segmentation

The computational demands of deep learning methods have largely restricted the size of the input images, and subdivision into patches has been the most popular workaround for processing larger images such as MRI volumes. For WML segmentation, a patch-based strategy can even have some benefits such as the ability to selectively sample more representative regions. However, most patch-based methods are inefficient in that they perform many redundant computations in the overlapping regions of neighboring patches. While some methods have been proposed for speeding up patch-based networks (eg, Li et al., 2014, as used by Vaidya et al., 2015), some recent segmentation approaches have used fully convolutional networks (FCNs; Long et al., 2015), which only contain layers that can be framed as convolutions (eg, pooling and up sampling), to perform dense prediction by producing segmented output that is of the same dimensions as the original images. Brosch et al. (2015) proposed a 3D FCN to process entire MRI volumes for MS WML segmentation. The network used a convolutional layer with 32 ($9 \times 9 \times 5$) filters to extract features from the input layer at each voxel location, and a deconvolutional layer that used the extracted features to predict a lesion mask and thereby classify each voxel of the image in a single operation. The authors called this model a convolutional encoder network due to its similarity to a convolutional autoencoder, and applied an efficient Fourier-based training algorithm (Brosch and Tam, 2015) to perform end-to-end training, which enabled feature learning to be driven by segmentation performance. By processing entire MRI volumes instead of patches, the algorithm avoids redundant calculations, and therefore could scale up more efficiently with image resolution. To overcome the problem of unbalanced classes without selective voxel sampling, the authors proposed a new objective function based on a weighted combination of sensitivity and specificity ($\approx$1:10 ratio), reformulated to be error terms that allowed for stable gradient computations. Optimization was performed with SGD. The method was evaluated on a large dataset of PDw and T2w volumes from an MS clinical trial, acquired from 45 different scanning sites, of 500 subjects that the authors split equally into training and test sets. By varying the training sample size, the authors showed that approximately 100 scans were sufficient for this framework to learn to segment the test scans optimally. More recent work by the authors (Brosch et al., 2016) has shown that adding more layers can further improve segmentation performance. Overall, the FCN approach applied to full MRI volumes can be seen as a promising alternative to patch-based methods, especially where computational efficiency is a concern.

3.3.3 MODELING DISEASE VARIABILITY

Changes in brain morphology and white matter lesions are two hallmarks of MS pathology, but their variability beyond volumetrics is poorly characterized. To further the understanding of complex MS pathology, Brosch et al. (2014) proposed using DBNs to build a statistical model of brain images that can automatically discover spatial patterns of variability in brain morphology and lesion distribution. The test

data was composed of MRIs of 474 MS patients, with each having a multimodal set of T1w, T2w, and PDw volumes with a resolution of $256 \times 256 \times 50$ voxels and a voxel size of 0.937 mm × 0.937 mm × 3.000 mm. In contrast to other methods for manifold learning, the DBN approach, with its capability for automatic feature learning, does not require a prebuilt proximity graph, which is particularly advantageous for modeling sparse and pseudorandom content such as lesions, because defining a suitable distance measure between lesion images would be very challenging. The proposed network consisted of a morphology DBN, a lesion DBN, and a joint DBN that modeled concurring morphological and lesion patterns. The input to the morphology DBN was a set of deformation fields computed by nonlinear registration of the T1w MRIs to a standard template. The input to the lesion DBN was a set of binary lesion masks (produced from the T2w and PDw scans) with the same transformations applied. Both the morphology and lesion DBNs were composed of three strided convolutional RBMs (sconvRBMs) and two dense RBMs with 16 and 2 hidden units. For the morphology DBN, the three sconvRBMs had stride sizes of $2 \times 2 \times 1$, $2 \times 2 \times 2$, $1 \times 1 \times 1$, filter sizes of $10 \times 10 \times 7$, $10 \times 10 \times 10$, $3 \times 5 \times 3$, and 32, 64, 32 filters, respectively. For the lesion DBN, the three sconvRBMs had stride sizes of $4 \times 4 \times 2$, $2 \times 2 \times 2$, $2 \times 2 \times 2$, filter sizes of $20 \times 20 \times 10$, $14 \times 14 \times 10$, $10 \times 14 \times 6$, and 32, 64, 64 filters, respectively. The joint DBN consisted of two pathways, each consisting of the first four layers of the morphology and lesion DBNs, respectively, and a fifth RBM layer with four hidden units, which replaced the fifth layer of the individual DBNs and combined the hidden unit activations of the fourth layer RBMs. Fig. 3.4 shows some images sampled from the learned manifolds. The results allowed the authors to make three main observations: (1) the model automatically discovered the classic patterns of MS pathology, such as enlarged ventricles and increased preventricular lesion load (Traboulsee and Li, 2008), as well as the more subtle ones, such as lesion load in specific structures (eg, brain stem); (2) the parameters of the joint model correlated stronger with MS clinical scores than the parameters of either individual model; and (3) the parameters of the individual models and the joint model correlated stronger with MS clinical scores than the traditional imaging biomarkers of brain volume and lesion volume. Overall, this study demonstrated that deep learning can be used to learn complex and clinically relevant features from brain images of MS patients, with very few assumptions by the user.

3.4 FUTURE RESEARCH NEEDS

We have reached an exciting time for deep learning, a field that is progressing very rapidly in academia, and also having well-publicized success in industry. For the neuroimaging researcher interested in applying deep learning, it seems almost impossible to keep up with the latest technical developments published in machine learning conferences and journals. However, it should be realized that despite research over the last several years that has produced very promising results and even

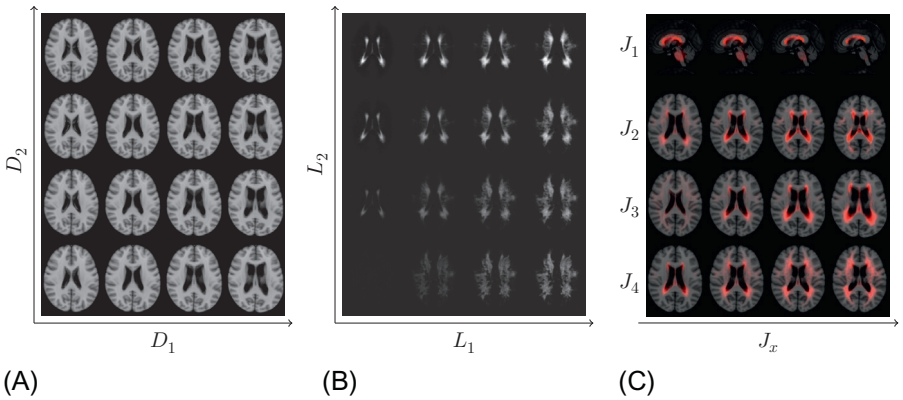

FIG. 3.4

Slices from generated volumes from the (A) morphology, (B) lesion, and (C) joint models. The morphology model captured ventricular enlargement (D_1) and decrease in brain size (D_2) as the main modes of variation. For the lesion model, L_1 captured an increase in lesion load throughout the WM, while L_2 captured primarily periventricular lesion load variations. The parameters of the joint model captured combinations of the variability found in the individual models.

some true breakthroughs, deep learning remains largely unvalidated for automated feature learning of brain images. This is because much of the work done by the machine learning community has been applied to much larger datasets of images with much lower dimensionality. This is evident, for example, from the fact that support for 3D convolutions has just recently been added to GPU libraries. Neuroimaging data has unique challenges, and much work still needs to be done to establish standards for designing and training deep networks, even just the basic ones described in this chapter. For example, determining the optimal number of layers, filter sizes, learning rates, and regularization strategy all need further investigation. Fortunately, there has been enough success so far to establish research momentum, and positive results will continue to come, and hopefully the advantages of deep learning seen in other fields will also be realized in neuroimaging. What we see as particularly promising are highly efficient models such as fully convolutional networks, because they allow full images to be processed, which facilitates the use of deeper models and larger datasets, as well as the experimentation with different network architectures.

ACKNOWLEDGMENTS

This work was supported by the Natural Sciences and Engineering Research Council of Canada, the Milan and Maureen Ilich Foundation, and the UBC Engineers-in-Scrubs Program. The authors gratefully acknowledge the valuable feedback from Drs. David Li and Anthony Traboulsee.

REFERENCES

Alonso-Ortiz, E., Levesque, I.R., Pike, G.B., 2015. MRI-based myelin water imaging: a technical review. Magn. Reson. Med. 73 (1), 70–81.

Barkhof, F., 2002. The clinico-radiological paradox in multiple sclerosis revisited. Curr. Opin. Neurol. 15 (3), 239–245.

Breiman, L., 2001a. Random forests. Mach. Learn. 45 (1), 5–32.

Breiman, L., 2001b. Statistical modeling: the two cultures. Stat. Sci. 16 (3), 199–231.

Brosch, T., Tam, R., 2015. Efficient training of convolutional deep belief networks in the frequency domain for application to high-resolution 2D and 3D images. Neural Comput. 27 (1), 211–227.

Brosch, T., Yoo, Y., Traboulsee, A., Li, D., Tam, R., 2014. Modeling the variability in brain morphology and lesion distribution in multiple sclerosis by deep learning. In: Proceedings of Medical Image Computing and Computer Assisted Intervention (MICCAI) Part II, pp. 463–470.

Brosch, T., Yoo, Y., Tang, L., Traboulsee, A., Li, D., Tam, R., 2015. Deep convolutional encoder networks for multiple sclerosis lesion segmentation. In: Proceedings of Medical Image Computing and Computer Assisted Intervention (MICCAI) Part III, pp. 3–11.

Brosch, T., Tang, L.Y.W., Yoo, Y., Li, D.K.B., Traboulsee, A., Tam, R., 2016. Deep 3D convolutional encoder networks with shortcuts for multiscale feature integration applied to multiple sclerosis lesion segmentation. In: IEEE Transactions on Medical Imaging, Special Issue on Deep Learning (in press).

Chiaravalloti, N.D., DeLuca, J., 2008. Cognitive impairment in multiple sclerosis. Lancet Neurol. 7 (12), 1139–1151.

Cho, K., Ilin, A., Raiko, T., 2011. Improved learning of Gaussian-Bernoulli restricted Boltzmann machines. In: Artificial Neural Networks and Machine Learning—ICANN 2011. Springer, New York, pp. 10–17.

Ciresan, D., Giusti, A., Schmidhuber, J., 2012. Deep neural networks segment neuronal membranes in electron microscopy images. In: Advances in Neural Information Processing Systems, pp. 1–9.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20 (3), 273–297.

Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 886–893.

Dauphin, Y.N., de Vries, H., Chung, J., Bengio, Y., 2015. RMSProp and equilibrated adaptive learning rates for non-convex optimization. arXiv 1502.04390v1.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. ImageNet: a large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, Piscataway, NJ, pp. 248–255.

Duchi, J., Hazan, E., Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. 12, 2121–2159.

Farley, B., Clark, W., 1954. Simulation of self-organizing systems by digital computer. Trans. IRE Prof. Group Inform. Theory 4 (4), 76–84.

Freund, Y., Haussler, D., 1992. Unsupervised learning of distributions on binary vectors using two layer networks. In: Proceedings of Advances in Neural Information Processing Systems, pp. 912–919.

Fukushima, K., 1980. Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol. Cybern. 36 (4), 193–202.

García-Lorenzo, D., Francis, S., Narayanan, S., Arnold, D.L., Collins, D.L., 2013. Review of automatic segmentation methods of multiple sclerosis white matter lesions on conventional magnetic resonance imaging. Med. Image Anal. 17 (1), 1–18.

Ghafoorian, M., Platel, B., 2015. Convolutional neural networks for MS lesion segmentation, method description of Diag team. In: Proceedings of IEEE International Symposium on Biomedical Imaging (ISBI): Grand Challenge in Longitudinal Multiple Sclerosis Lesion Segmentation.

Gönen, M., Alpaydın, E., 2011. Multiple kernel learning algorithms. J. Mach. Learn. Res. 12, 2211–2268.

Guo, Y., Wu, G., Commander, L.A., Szary, S., Jewells, V., Lin, W., Shen, D., 2014. Segmenting hippocampus from infant brains by sparse patch matching with deep-learned features. In: Medical Image Computing and Computer-Assisted Intervention—MICCAI 2014. Springer, New York, pp. 308–315.

Haacke, E.M., Xu, Y., Cheng, Y.C.N., Reichenbach, J.R., 2004. Susceptibility weighted imaging (SWI). Magn. Reson. Med. 52 (3), 612–618.

Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., Pal, C., Jodoin, P.M., Larochelle, H., 2015. Brain tumor segmentation with deep neural networks. arXiv 1505.03540.

Hinton, G.E., 2002. Training products of experts by minimizing contrastive divergence. Neural Comput. 14 (8), 1771–1800.

Hinton, G.E., 2010. A practical guide to training restricted Boltzmann machines. Momentum 9 (1), 926.

Hinton, G.E., Salakhutdinov, R., 2006. Reducing the dimensionality of data with neural networks. Science 313 (5786), 504–507.

Hinton, G.E., Osindero, S., Teh, Y.W., 2006. A fast learning algorithm for deep belief nets. Neural Comput. 18 (7), 1527–1554.

Hjelm, R.D., Calhoun, V.D., Salakhutdinov, R., Allen, E.A., Adali, T., Plis, S.M., 2014. Restricted Boltzmann machines for neuroimaging: an application in identifying intrinsic networks. NeuroImage 96, 245–260.

Hubel, D.H., Wiesel, T.N., 1962. Receptive fields, and binocular interaction and functional architecture in the cat's visual cortex. J. Physiol. 160 (1), 106.

Hubel, D.H., Wiesel, T.N., 1968. Receptive fields and functional architecture of monkey striate cortex. J. Physiol. 195 (1), 215–243.

Jain, A.K., Farrokhnia, F., 1990. Unsupervised texture segmentation using Gabor filters. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, pp. 14–19.

Kamnitsas, K., Chen, L., Ledig, C., Rueckert, D., Glocker, B., 2015. Multi-scale 3D convolutional neural networks for lesion segmentation in brain MRI. In: Proceedings of Ischemic Stroke Lesion Segmentation Challenge, pp. 13–16.

Kim, M., Wu, G., Shen, D., 2013. Unsupervised deep learning for hippocampus segmentation in 7.0 tesla MR images. In: Proceedings of Medical Image Computing and Computer Assisted Intervention (MICCAI) Machine Learning in Medical Imaging (MLMI) Workshop, pp. 1–8.

Kingma, D., Ba, J., 2014. Adam: a method for stochastic optimization. arXiv 1412.6980.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Proceedings of Advances in Neural Information Processing Systems, pp. 1097–1105.

Kurtzke, J.F., 1983. Rating neurologic impairment in multiple sclerosis: an expanded disability status scale (EDSS). Neurology 33 (11), 1444–1452.

Laule, C., Vavasour, I.M., Moore, G.R.W., Oger, J., Li, D.K.B., Paty, D.W., MacKay, A.L., 2004. Water content and myelin water fraction in multiple sclerosis. A T2 relaxation study. J. Neurol. 251 (3), 284–293.

Le Bihan, D., Mangin, J.F., Poupon, C., Clark, C.A., Pappata, S., Molko, N., Chabriat, H., 2001. Diffusion tensor imaging: concepts and applications. J. Mag. Reson. Imaging 13 (4), 534–546.

LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D., 1989. Backpropagation applied to handwritten zip code recognition. Neural Comput. 1 (4), 541–551.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86 (11), 2278–2324.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep Learning. Nature 521 (7553), 436–444.

Lee, H., Grosse, R., Ranganath, R., Ng, A.Y., 2009. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 609–616.

Lee, H., Grosse, R., Ranganath, R., Ng, A.Y., 2011. Unsupervised learning of hierarchical representations with convolutional deep belief networks. Commun. ACM 54 (10), 95–103.

Li, H., Zhao, R., Wang, X., 2014. Highly efficient forward and backward propagation of convolutional neural networks for pixelwise classification. arXiv 1412.4526 [cs].

Liu, S., Cai, W., Che, H., Pujol, S., Kikinis, R., Feng, D., Fulham, M., 2014a. Multi-modal neuroimaging feature learning for multi-class diagnosis of Alzheimer's disease. IEEE Trans. Biomed. Eng. 62 (4), 1132–1140.

Liu, S., Liu, S., Cai, W., Che, H., Pujol, S., Kikinis, R., Fulham, M., Feng, D., 2014b. High-level feature based PET image retrieval with deep learning architecture. J. Nucl. Med. 55 (Suppl. 1), 2028–2028.

Lladó, X., Oliver, A., Cabezas, M., Freixenet, J., Vilanova, J.C., Quiles, A., Valls, L., Ramió-Torrentà, L., Àlex Rovira, 2012. Segmentation of multiple sclerosis lesions in brain MRI: a review of automated approaches. Inform. Sci. 186 (1), 164–185.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), pp. 3431–3440.

Lowe, D.G., 1999. Object recognition from local scale-invariant features. In: IEEE International Conference on Proceedings of Computer Vision, vol. 2, pp. 1150–1157.

Mallat, S.G., 1989. A theory for multiresolution signal decomposition: the wavelet representation. IEEE Trans. Pattern Anal. Mach. Intell. 11 (7), 674–693.

Nair, V., Hinton, G.E., 2010. Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning, pp. 807–814.

Ojala, T., Pietikainen, M., Maenpaa, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. 24 (7), 971–987.

Plis, S.M., Hjelm, D.R., Salakhutdinov, R., Allen, E.A., Bockholt, H.J., Long, J.D., Johnson, H.J., Paulsen, J.S., Turner, J.A., Calhoun, V.D., 2014. Deep learning for neuroimaging: a validation study. Front. Neurosci. 8, Article 229.

Polyak, B.T., Juditsky, A.B., 1992. Acceleration of stochastic approximation by averaging. SIAM J. Control Optim. 30 (4), 838–855.

Raina, R., Madhavan, A., Ng, A.Y., 2009. Large-scale deep unsupervised learning using graphics processors. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 873–880.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. Nature 323, 533–536.

Sainath, T.N., Mohamed, A.R., Kingsbury, B., Ramabhadran, B., 2013. Deep convolutional neural networks for LVCSR. In: Proceedings of 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, Piscataway, NJ, pp. 8614–8618.

Salakhutdinov, R., Hinton, G., 2012. An efficient learning procedure for deep Boltzmann machines. Neural Comput. 24 (8), 1967–2006.

Scherer, D., Müller, A., Behnke, S., 2010. Evaluation of pooling operations in convolutional architectures for object recognition. In: Artificial Neural Networks-ICANN 2010. Springer, New York, pp. 92–101.

Shen, L., Kim, S., Qi, Y., Inlow, M., Swaminathan, S., Nho, K., Wan, J., Risacher, S.L., Shaw, L.M., Trojanowski, J.Q., et al., 2011. Identifying neuroimaging and proteomic biomarkers for MCI and AD via the elastic net. In: Multimodal Brain Image Analysis. Springer, New York, pp. 27–34.

Song, A.W., Huettel, S.A., McCarthy, G., 2006. Functional neuroimaging: basic principles of functional MRI. In: Cabeza, R., Kingstone, A. (Eds.), Handbook of Functional Neuroimaging of Cognition, Second ed. MIT Press, Cambridge, MA, pp. 21–52.

Suk, H.I., Lee, S.W., Shen, D., Alzheimer's Disease Neuroimaging Initiative, 2014. Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis. NeuroImage 101, 569–582.

Suk, H.I., Lee, S.W., Shen, D., Alzheimer's Disease Neuroimaging Initiative, 2015. Latent feature representation with stacked auto-encoder for AD/MCI diagnosis. Brain Struct. Funct. 220 (2), 841–859.

Tenenbaum, J.B., De Silva, V., Langford, J.C., 2000. A global geometric framework for nonlinear dimensionality reduction. Science 290 (5500), 2319–2323.

Tieleman, T., 2008. Training restricted Boltzmann machines using approximations to the likelihood gradient. In: Proceedings of the 25th International Conference on Machine Learning, pp. 1064–1071.

Tomassini, V., Matthews, P.M., Thompson, A.J., Fuglø, D., Geurts, J.J., Johansen-Berg, H., Jones, D.K., Rocca, M.A., Wise, R.G., Barkhof, F., Palace, J., 2012. Neuroplasticity and functional recovery in multiple sclerosis. Nat. Rev. Neurol. 8 (11), 635–646.

Traboulsee, A., Li, D.K., 2008. Conventional MR imaging. Neuroimaging Clin. N. Am. 18 (4), 651–673.

Vaidya, S., Chunduru, A., Muthuganapathy, R., Krishnamurthi, G., 2015. Longitudinal multiple sclerosis lesion segmentation using 3D convolutional neural networks. In: Proceedings of IEEE International Symposium on Biomedical Imaging (ISBI): Grand Challenge in Longitudinal Multiple Sclerosis Lesion Segmentation.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A., 2010. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. 11, 3371–3408.

Werbos, P., 1974. Beyond regression: new tools for prediction and analysis in the behavioral sciences. Ph.D. thesis, Harvard University, Cambridge, MA.

Wu, G., Kim, M., Wang, Q., Gao, Y., Liao, S., Shen, D., 2013. Unsupervised deep feature learning for deformable registration of MR brain images. In: Medical Image Computing and Computer-Assisted Intervention—MICCAI 2013. Springer, pp. 649–656.

Yoo, Y., Brosch, T., Traboulsee, A., Li, D., Tam, R., 2014. Deep learning of image features from unlabeled data for multiple sclerosis lesion segmentation. In: Proceedings of Medical Image Computing and Computer Assisted Intervention (MICCAI) Machine Learning in Medical Imaging (MLMI) Workshop, pp. 117–124.

Yuan, M., Lin, Y., 2006. Model selection and estimation in regression with grouped variables. J. R. Stat. Soc. B 68 (1), 49–67.

Zeiler, M.D., 2012. ADADELTA: an adaptive learning rate method. arXiv 1212.5701.

Zhang, W., Li, R., Deng, H., Wang, L., Lin, W., Ji, S., Shen, D., 2015. Deep convolutional neural networks for multi-modality isointense infant brain image segmentation. NeuroImage 108, 214–224.