**Introduction**

The world population is about to face an explosion in its elderly population, with the number of elderly in the world doubling, but the total population slightly decreasing by 2050 (Katzman and Fox). This means the population of the world’s elderly suffering from various forms of dementia, such as Alzheimer’s, will double as well (Katzman and Fox). Because of this, the cost of dementia to society will increase eight fold in the coming thirty-five years, making it imperative that dementia research, especially for Alzheimer’s, is significantly improved (Katzman and Fox).

One important barrier to the development of Alzheimer’s research is the ability to diagnose Alzheimer’s early and accurately in order to ensure safe and effective research of new treatments. However, current methods of diagnosis for Alzheimer's disease can only be certain when the disease has already progressed to an advanced stage of dementia (McKhann et al). Furthermore, early diagnosis methods such as lumbar puncture (Menéndez-González) or a MRI scan (Glover) are invasive, expensive, and may have side effects, which make diagnosis less accessible to people who may be at risk for developing the disease (Glover).

Diagnosis through cognitive testing provides an inexpensive, non-invasive alternative. Unfortunately, it is very hard to use these methods to diagnose Alzheimer's early because Alzheimer’s symptoms are similar to other causes of dementia. Definitive diagnosis can only technically be given after a post-mortem exam (Walker et al). Developing a more affordable, less invasive diagnosis method that is sensitive to early indicators is essential to the progress of Alzheimer's research by allowing researchers to study trends in more detail from an early stage of the disease (Menéndez-González).

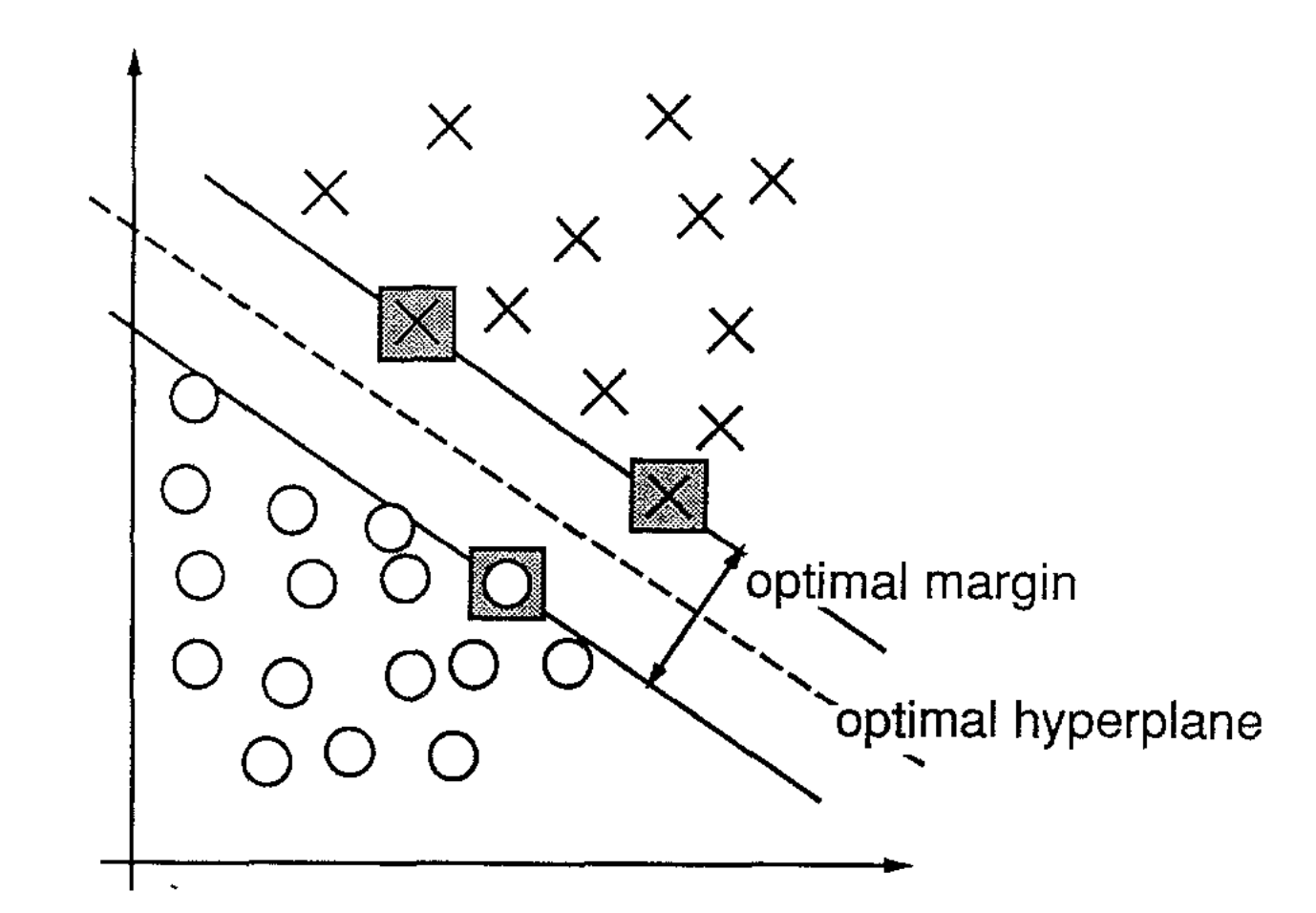
This project aims to use a machine learning architecture called artificial neural networks to identify and utilize some of these early indicators found through cognitive testing. Artificial neural networks (ANN or neural nets) are algorithms that are used to find complex relationships in large datasets (Jain et al). However, any single person did not conceptualize neural nets. The type of the neural net implementation in this project is similar to the conception of James Anderson or Teuvo Kohonen. Neural nets have many applications. Neural nets are used in stock market analysis, image and voice recognition, and recently, medical diagnosis (Basu et al). This is why neural nets may be a good method to give earlier diagnoses, especially as it pertains to non-invasive cognitive tests, because they are able to identify complex patterns in the data.

Previous attempts to use neural nets in Alzheimer’s diagnosis have used data from SPECT (single-photon emission computerized tomography) scans and achieve an area of .91 under an ROC (Receiver Operating Characteristic) curve (Page et al). Other studies have been able to distinguish normal patients from those with mild cognitive impairment using fuzzy neural networks, which although useful, does not indicate the presence of Alzheimer's (Anand et al). Mild cognitive impairment only tells us that the individual has started to develop some cognitive problems but does not specify if the cause is Alzheimer’s. One study with a very small test set of only 27 sample cases (test subjects) achieved 100% accuracy using probabilistic neural nets (Sankari and Adeli).

This study aims to achieve a mean squared error less than 0.1 on over 34,000 test cases using non-invasive cognitive testing by using deep neural networks, support vector machines, and k-means clustering trained over the remaining 34,000 cases in the National Alzheimer's Coordinating Center's (NACC) Uniform Dataset (UDS).

**Support Vector Machines**

Support Vector Machines (SVMs) have successfully classified genetic mutations (Sehgal, Gondal, and Dooley, 2004), tracked moving objects (Tian et al., 2007), and improved training methods for image classification (Foody and Mathur, 2004). SVMs are another machine learning algorithm that can perform both regression and classification (Cortes and Vapnik, 1995). They are generally supervised learning algorithms, however, they can also perform unsupervised learning (Ben-Hur, 2008). An SVM creates a hyperplane with the largest margin between two classes in a non-linear space (Cortes and Vapnik, 1995).

****

*Fig. 1: Example SVM Classification (Cortes and Vapnik, 1995)*

Fig. 1 (above), is an example of an SVM solving a 2 dimensional classification problem. Here, each input vector is an O or an X, depending on its classification. The boxed data points are support vectors (Cortes and Vapnik, 1995): support vectors are data points that lie on the edge of the division between classes (Cortes and Vapnik, 1995).

Unfortunately, this idea breaks if the data is not linearly separable, like the XOR logical function (Winston, 2014). This is where the vector function comes into play - it transforms the data into a higher-dimensional space where it is separable by a single hyperplane (Cortes and Vapnik, 1995).

In an SVM, this hyperplane minimizes the expression on top for values constrained by the inequality on the second line in fig. 2 (Cortes and Vapnik, 1995).

and is of the same sign as

*Fig. 2: Constraints on an SVM*

In *fig. 2*, is a vector normal to the decision boundary, is a member of one of the classes, and is some scalar (Bennet and Campbell, 2000). is minimized because the combined width of the margins is inversely proportional to . Thus, maximizing it yields the optimal decision boundary (Winston, 2014). Putting into a quadratic form is a mathematical convenience as it sets up a convex learning space (Winston, 2014).

After using Lagrangian multipliers to reconcile the expression to be minimized and its constraints, the maximization of the margin’s width depends on the dot product of the support vectors (Winston, 2014). Kernel functions solve the dimensionality problem for this operation by describing the dot product in terms of the original space (Winston, 2014). Thus, maximizing the kernel function maximizes the margins between the support vectors (Steinwart and Christmann, 2008), which make SVMs very powerful tools and allow the efficient use of conventional optimization algorithms for learning.

**Artificial Neural Networks**

Artificial neural networks (neural nets) are used to find complex relationships in large datasets (Jain et al., 1996). They have seen success in many applications, such as image and voice recognition, robot control systems, and medical diagnosis (Basu et al., 2010).

Neural nets are analogous to the structure of neurons in the brain (Roberts, 2000): like neurons receive a signal, transform it, and pass it to the next neuron in the chain, each neuron in the net takes the sum of its inputs, applies an activation function to it, and passes the output along synapses (Nielson, 2015). These synapses have weights that change as the neural net learns (Nielson, 2015).

The input to a neuron is a collection of all of the “pre-activation” values. Pre-activation values are the products of the weights of the synapse and its input (Ng, no date). The sum of these pre-activation values is passed to an activation function, , which produces the output of the neuron (). Fig. 4 is the structure of a neural net: circles are neurons and arrows are synapses.

Input Layer

Hidden Layer

Output Layer



*Fig. 4: Basic Neural Net Architecture*

However, this architecture still needs some mechanism that allows it to “learn”. This is done using an optimization algorithm, such as gradient descent with backward propagation of errors (backprop) (Nielson, 2015).

**Deep Neural Networks**

Deep Neural Networks (DNNs) are feed forward neural nets with multiple hidden layers (Hinton et al., 2012). DNNs are able to use many hidden layers with non-linear activations to more effectively learn abstract relationships among the input features (Hinton et al., 2012). Just like neural nets, DNNs are trained using backprop with relation to different loss functions depending on the learning problem; common loss functions are an l2-loss (for regression) and cross-entropy (for classification) (Hinton et al. 2012).

Like normal neural nets, DNNs have seen much success in handwritten digit recognition (Cireşan et al., 2010), breast cancer diagnosis (Litgens et al., 2016), image classification (Cireşan et al., 2012), and speech recognition (Hinton et al. 2012).

**Material and Methods**

**Use of the National Alzheimer’s Coordinating Council Universal Dataset**

The data used in this project to train the neural network was obtained from a database of the NACC (National Alzheimer’s Coordinating Council) through the normal procurement process, more of which is detailed on the NACC website. The data was from the UDS (Uniform Data Set) version 2, and was cleaned up to make it compatible for use with a neural network. There were many problems with the original data. The first and most important problem was the lack of uniformity of data entries for the various tests that were part of the UDS. The data set originally contained ninety-eight thousand visits with four hundred data values recorded for each person after every visit. These tests, or data categories, included basic demographics, such as age, sex, etc. as well as specific cognitive tests, motor skills assessments, etc. However, many times, a data value was not available for a patient because the visit was a follow-up over phone, or they did not speak English, etc. In these cases, their data would not provide a complete input vector so it was eliminated from the final data set. In addition many tests did not contain enough data, i.e. not enough patients were able to complete the test, or it had data that could not be read by the neural net, such as a description instead of a numerical value. So once again, these data values had to be deleted.

To cleanup the data, first, all of the data categories and a list of all of the possible values they contained were made. Then all categories that required written explanation (those whose code ended in X) were deleted. Then all data values such as home address, which was unnecessary, or any that were redundant, such as multiple entries for age and date of birth, were deleted so that only one entry remained. The next step was to recode some of the keys for the categories to be more compatible with the numerical nature of a neural net. In order to do this, the various symptoms were assigned numerical values which were arranged to reflect severity.

In addition to data from the tests, the dataset included the clinical diagnosis of the patient. The neural net used this clinical diagnosis as the correct output.

**Macros Used**

In order to automate the process of cleaning up this huge data file, multiple macros were written in VBA (visual basic for applications). These macros were written as part of a copy of the original CSV (comma separated value) file that was saved as a .xlsm (macro enabled excel) file.

The first macro created a list of the possible (unique) values of each variable in the dataset. It then added a blank column next to the list of unique values to input the corresponding new value for each original unique value in the data set. In addition, each variable that was to be deleted was assigned “N/A” in the new key. Last, any data point that indicated that the patient that it pertained to would not have a complete set of data had “TAKE OUT” assigned to it in the new key. Some examples of this are if the patient didn’t take part in one of the oral tests. This generally indicated that the patient did not speak English and thus did not participate in any of the oral testing procedures.

The second macro then took the new keys and replaced all of the old data values from the original file with new values from the new key.

The third macro went through the data file and did any special replacements whose value not only had to be changed based on the new key but also the age or sex of the patient. For example, in order to allow the data from as many people as possible to be usable, any missing values for height were entered in as the average expected height for the patient depending on their age and sex. It also removed the data of all of the tests that had been assigned an “N/A” in first macro and deleted the data of all patients that had the “TAKE OUT” designation (also from the first macro). Fig. 4 (below) describes the use of the macros further and also shows screenshots of the excel sheet as it was being changed.

|  |  |
| --- | --- |
|  |  |

*Fig. 4 : Data Cleanup*

**Support Vector Machine Construction**

The support vector machine used in this project was implemented in SciKit Learn (SKLearn). SKLearn is a machine learning API that facilitates the implementation of machine learning algorithms. Out of the three Support Vector Regression (SVR) techniques supported by SKLearn, Nu Support Vector Classifier (NuSVC) was chosen as it performed best in initial testing.

For the NuSVC model, the final (6th out of 7) version used the “poly” kernel function. The other options were also tried, but they did not perform as well in testing. The value for gamma, which influences how much the model overfits (Cortes and Vapnik, 1995), was also changed and proved to have a significant impact on the model.

NuSVC used a hinge loss and nu regression instead of epsilon regression. Epsilon regression uses as the size of the margin within which all data points should be contained to construct the loss function (Smola and Schölkopf, 2004). Instead, nu regression uses as an upper bound on the fraction of training examples outside the margin and as a lower bound on the number of support vectors (Chang and Lin, 2009).

**Deep Neural Net Construction**

The first DNN had 8 hidden layers with 30 units per layer and an exponential linear unit (ELU) activation function for all of the layers. ELU alleviates the bias shift problem of the more popular ReLU (rectified linear unit) while retaining ReLU’s ability to avoid the vanishing gradient problem (Clevert, Unterthiner, and Hochreiter, 2016). In addition, ELU is a zero-centered function, which has been shown to speed up learning in neural nets (LeCun, Kanter, and Solla, 1991).

Unfortunately, this setup did not seem to be learning well. Thus, over the next ten variations of the network, the neural net developed an architecture with 10 hidden layers and used combinations of ELU, ReLU, TanH, Softmax, Softplus, and Sigmoid as the activation functions in the network. Lastly, the final neural net used gradient descent to optimize the parameters. Even though gradient descent is slower, it was more accurate than Adagrad and Adam.

Tensorflow, which is an open source machine learning library released by Google, was used to construct and train the neural net as it allows for fast and highly parallelizable execution of machine learning tasks (Abadi et al., 2015).

**Proofs of Concept**

There were four proofs of concept used to check the validity of the algorithm’s operation. These were simple relationships that could be checked manually to prevent any error in the programming from skewing the check of the algorithms’ functioning.

First they were checked against a data set of 5 examples for which there were 10 components each. In this test, the data values were either all ones or all zeros. In the case that they were all ones, the correct output was 1 and in the case that they were all zeros, the correct output was zero as well.

The second test had the same dimensions with respect to the data set, however no matter what the input values were, the output was always 1.

The third test had the same dimensions once more. However, this time, the input values were between positive and negative 100 and contained decimal values. The function that was modeled was dot multiplication of the input array with a randomly constructed matrix of the appropriate size to produce a one by one matrix, or a single decimal value as the correct output.

The final test had dimension that were one fourth of the data set that was to be used for the actual experimentation and modeled the same dot multiplication function as the previous proof of concept.

Only after the neural net was able to successfully pass all of these proofs of concepts was final testing started.

**Bibliography**

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S.,

Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X. and Google Research (2015) ‘TensorFlow: Large-scale machine learning on heterogeneous distributed systems’, .

Anand, Mahesh S., Mukunda M. Rao, Shyam N. Prabhu, Samraj D. Simeon, D.

Karthikeyan, and Rashmi Snigdha. "Automated Diagnosis of Early Alzheimer's Disease Using Fuzzy Neural Network" Springer Link. Springer, 27 Nov. 2008. Web. 27 Oct. 2015.

Basu, Jayantha Kumar, Debnath Bhattacharyya, and Tai Hoon Kim. "The Use of

Artificial Neural Networks in Pattern Recognition." International Journal of Software Engineering and Its Applications 4.2 (2010): n. pag. SERSC. SERSC, Apr. 2010. Web. 26 Oct. 2015.

Ben-Hur, A. (2008) ‘Support vector clustering’, Scholarpedia, 3(6), p. 5187. doi:

10.4249/scholarpedia.5187.

Bennett, K.P. and Campbell, C. (2000) ‘Support Vector Machines: Hype or Hallelujah?’,

SIGKDD Explorations, 2(2), pp. 2–13.

Chang, C.-C. and Lin, C.-J. (2009) Training ν-Support Vector Regression: Theory and

Algorithms, 22 July. .

Cires ̧an, D., Meier, U. and Schmidhuber, J. ̈rgen (2012) Multi-column Deep Neural

Networks for Image Classification. Available at: https://arxiv.org/pdf/1202.2745.pdf (Accessed: 26 December 2016).

Cireşan, D., Meier, U., Gambardella, L. and Schmidhuber, J. (2010) ‘Deep, big, simple

neural nets for handwritten digit recognition’, Neural computation., 22(12), pp. 3207–20.

Clevert, D.-A., Unterthiner, T. and Hochreiter, S. (2016) ‘FAST AND ACCURATE

DEEP NETWORK LEARNING BY EXPONENTIAL LINEAR UNITS (ELUS)’, ICLR: . Available at: https://arxiv.org/pdf/1511.07289.pdf (Accessed: 27 December 2016).

Cortes, C. and Vapnik, V. (1995) ‘Support-vector networks’, Machine Learning, 20(3),

pp. 273–297. doi: 10.1007/bf00994018.

Foody, G.M. and Mathur, A. (2004) ‘Toward intelligent training of supervised image

classifications: Directing training data acquisition for SVM classification’,

Remote Sensing of Environment, 93(s 1–2), pp. 107–117. doi: 10.1016/j.rse.2004.06.017

Glover, Lacie. "Why Does an MRI Cost So Darn Much?" *Time*. Money, 16 July 2015.

Web. 28 Jan. 2016.

Hornik, Kurt. "Approximation Capabilities of Multilayer Feedforward Networks."

*Approximation Capabilities of Multilayer Feedforward Networks*. Neural Networks, 19 Mar. 2003. Web. 06 Feb. 2016.

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.-R., Jaitly, N., Senior, A.,

Vanhoucke, V., Nguyen, P., Sainath, T. and Kingsbury, B. (2012) ‘Deep neural networks for acoustic modeling in speech recognition’, .

Jain, Anil K., Jianchang Mao. “Artificial Neural Networks: A Tutorial” Institute of

Electrical and Electronics Engineers. Computers Society, March 1996. Web. 18 Jan. 2016

Katzman, R., and PJ Fox. "Epidemiology of Alzheimer's Disease: From Gene to

Prevention." *Epidemiology of Alzheimer's Disease: From Gene to Prevention | Richard Mayeux | Springer*. Ed. Richard Mayeux. Springer Science and Business Media, 6 Dec. 2012. Web. 04 Feb. 2016.

Lecun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep Learning." *Nature* 521.7553

(2015): 436-44. Web. 7 Feb. 2016.

LeCun, Y., Kanter, I. and Solla, S.A. (1991) ‘Eigenvalues of covariance matrices:

Application to neural-network learning.’, Physical Review Letters, 66(18), pp. 2396–2399.

McKhann, Guy, David Drachman, Marshall Folstein, Robert Katzman, Donald Price, and

Emanuel Stadlan. "Clinical Diagnosis of Alzheimer's Disease." *Clinical Diagnosis of Alzheimer's Disease*. Neurology, July 1984. Web. 26 Jan. 2016.

Menéndez-González, Manuel. “Routine Lumbar Puncture for the Early Diagnosis of

Alzheimer’s Disease. Is It Safe?” *Frontiers in Aging Neuroscience* 6 (2014): 65. *PMC*. Web. 5 Feb. 2016.

MIT OpenCourseWare and Winston, P. (2014) 16. Learning: Support vector machines.

Available at: https://www.youtube.com/watch?v=\_PwhiWxHK8o (Accessed: 27 December 2016).

Montavon, G., Braun, M.L. and Müller, K.-R. (2011) ‘Kernel analysis of deep

networks’, Journal of Machine Learning Research, 12, pp. 2563–2581.

Mueller, Susanne G. et al. “Ways toward an Early Diagnosis in Alzheimer’s Disease: The

Alzheimer’s Disease Neuroimaging Initiative (ADNI).” *Alzheimer’s & dementia : the journal of the Alzheimer’s Association* 1.1 (2005): 55–66. *PMC*. Web. 6 Feb. 2016.

Ng, Andrew. “Lecture Notes #1” CS 229 Machine Learning Course Materials. nd. Web.

10 Feb. 2016 <http://cs229.stanford.edu/notes/cs229-notes1.ps>

Page, M. P., R. J. Howard, J. T. O'Brien, M. S. Buxton-Thomas, and A. D. Pickering.

"Use of neural networks in brain SPECT to diagnose Alzheimer's disease." National Center for Biotechnology Information. U.S. National Library of Medicine, 37 Feb. 1996. Web. 27 Oct. 2015.

Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. "Learning

Representations by Back-propagating Errors." *Nature* 323.6088 (1986): 533-36. Web. 4 Feb. 2016.

Sankari, Z., and H. Adeli. "Probabilistic Neural Networks for Diagnosis of Alzheimer's

Disease using Conventional and Wavelet Coherence." National Center for Biotechnology Information. U.S. National Library of Medicine, 15 Apr. 2011. Web. 27 Oct. 2015.

Sehgal, M.S.B., Gondal, I. and Dooley, L. (2004) ‘Statistical neural networks and support

vector machine for the classification of genetic mutations in ovarian cancer - IEEE Xplore document’, Computational Intelligence in Bioinformatics and Computational Biology, .

Steinwart, I. and Christmann, A. (2008) Support vector machines. New York: Springer-

Verlag New York.

Synman, Jan A. “Practical Mathematical Optimization: An Introduction to Basic

Optimization Theory and Classical and New Gradient-Based Algorithms.” Springer Publishing, nd. 2005. Web. 12 Feb. 2016

Tian, M., Zhang, W. and Liu, F. (no date) ‘On-line ensemble SVM for robust object

tracking’, Computer Vision – ACCV 2007, 4843, pp. 355–364. doi: 10.1007/978-3-540-76386-4\_33.

Walker, Z., D. C. Costa, R. WH Walker, K. Shaw, S. Gasinovic, G. Livingston, P. Ince, I.

G. McKeith, and C. LE Katona. "Differentiation of Dementia with Lewy Bodies from Alzheimer's Disease Using a Dopaminergic Presynaptic Ligand." Journal of Neurology, Neurosurgery, and Psychiatry, 10 Jan. 2002. Web. 1 Feb. 2016.