# Samouczący się system sterowania ogrzewaniem w domu inteligentnym

# Wstęp

Niniejsza prace poświeciłem zastosowaniu systemu ogrzewania w domu inteligentnym. Podzieliłem ja na 4 rozdziały. W pierwszym odniosłem źródeł, gdzie w odparta o naukowe definicje wyjaśniam najważniejsze pojęcia związane z praca. Np. pojęcie domu inteligentnego, cechy takiego domu, uczenie maszynowe. Przedstawiłem także podział algorytmów uczenia maszynowego ze względu na sposób działania. W tej części przedstawiłem także krótki rys historyczny domów inteligentnych i uczenia maszynowego.

W rozdziale drugim opisałem system, który napisałem w języku programowania C#. Napisany program ma rozpoznawać powrót właściciela domu i dostosowywać do niego prace urządzeń grzewczych. Jak również zastosowane biblioteki oraz wybrane algorytmy.

Kolejny rozdział opisuje szczegóły dotyczące obiektu symulacji i badań, jak również opis badan i wyniki symulacji.

Całość podsumowałem wnioskami wynikającymi z badań.

# Pojęcia związane z pracą

### 1.1 Pojęcie domu inteligentnego

"Inteligentny budynek to zespół jego możliwości technicznych pozwalających na podstawie stanów różnych czynników wewnętrznych i zewnętrznych odpowiednio reagować, w celu zapewnienia jego użytkownikom komfortowych warunków życia i wysokiego poziomu bezpieczeństwa [Jacek Włodarczyk, Zbigniew Podosek " Systemy Teletechniczne Budynków Inteligentnych"]

Intelligent Building Institute mówi, że inteligentny budynek jest budynkiem, który integruje różne systemy, aby skutecznie, w sposób skoordynowany zarządzać zasobami w celu zapewnienia jak najlepszego funkcjonowania jego użytkowników, maksymalizować oszczędności w zakresie inwestycji i kosztów operacyjnych oraz umożliwiać maksymalną elastyczność.[Intelligent Building Systems Albert Ting-pat So,Wai Lok Chan 1999 Springer US]

### 1.2.1 Historia domów inteligentnych

Pojęcie Intelligent Building Systems pojawiło się latach 90-tych ubiegłego stulecia w Stanach Zjednoczonych. W dużych zakładach produkujących samoloty w procesach o wysokim stopniu automatyzacji i robotyzacji pojawiły się problemy zarządzania i sterowania halami produkcyjnymi, ich wyposażeniem i zmiennością funkcji. Dostosowanie pomieszczeń fabrycznych i biuro-projektowych do zmieniającej się dynamicznie produkcji wymagało olbrzymiego nakładu sił i środków. Coraz częściej obiekty te posiadały bardzo rozwinięta infrastrukturę informatyczną i koszty ich eksploatacji zaczęły stanowić poważny udział w kosztach całego przedsięwzięcia produkcyjnego. Pojawiła się zatem potrzeba globalnego spojrzenia na te zasoby jakimi są hale produkcyjne oraz pomieszczenia biurowo-projektowe. Szybko okazało się, że nie można przenieść wprost doświadczeń z zakresu automatyzacji i robotyzacji produkcji na te obiekty. Wówczas pojawiło się pojęcie inteligentnych systemów budynków (Intelligent Building Sysytems) nie do końca na początku rozumiane jednoznacznie, ale intuicyjnie chodziło o wykorzystanie metod i narzędzi informatycznych do optymalizacji eksploatacji tych obiektów w takim stopniu, aby działanie to wspierało procesy jakie odbywają się w tych pomieszczeniach.

### 1.2.2 Cechy inteligentnego budynku

1. Wysoki poziom estetyki budynku

2. Wysoki poziom automatyzacji prac biurowych

3. wykorzystanie zaawansowanych technik telekomunikacyjnych

4. Automatyczna eksploatacja systemów budynku

5. Możliwość dokonywania szybkich zmian lokalizacyjnych wewnątrz budynku

Uzyskanie takich cech jest możliwe jedynie przy wyposażeniu budynku w określone elementy realizujące zaprogramowane funkcje, takie jak: 1. Inteligentne sensory 2. Jednostki sterujące 3. Jednostka nadzorująca 4. Okablowanie strukturalne Dwie pierwsze grupy elementów są wspaniałym przykładem zastosowania systemów wbudowanych i na nich skupimy nasza uwagę. Funkcję jednostki nadzorującej pełni zazwyczaj komputer stacjonarny umieszczony w centrum zarządzania budynkiem (jeśli jest to obiekt duży) lub mikrokontroler umieszczony w szafie sterowniczej budynku jeśli odnosimy to do domu mieszkalnego. Okablowanie strukturalne stanowi nieodzowny element zapewniający komunikację w obiekcie.

### 1.2. Uczenie maszynowe

### 1.2.1 Definicja uczenia maszynowe

Mówimy, że maszyna uczy się zadania T w oparciu o doświadczenie E i miarę jakości P, jeśli wraz z przyrostem doświadczenia E poprawia się jakość wykonywanego zadania T mierzona przez miarę P. [T. M. Mitchell. Machine Learning. McGraw Hill, 1997.]

Uczenie - zmiany w systemie adaptującym się pozwalające mu w przyszłości działać

bardziej efektywnie na takich samych zadaniach lub zadaniach o podobnych charakterze [Simon 1983]

### 1.2.2 Historia uczenia maszynowego

### 1.2.3 Rodzaje algorytmów uczenia maszynowego

Algorytmy uczenia maszynowego możemy pogrupować wg dwóch kryteriów:

1. Ze względu na styl uczenia
2. Ze względu na sposób działania

Biorąc pod uwagę styl uczenia wyróżniamy:

* algorytmy nadzorowane
* algorytmy bez nadzoru
* algorytmy z częściowym nadzorem

**Algorytmy nadzorowane** sposób uczenia maszynowego, które zakłada obecność ludzkiego nadzoru nad tworzeniem funkcji odwzorowującej wejście systemu na jego wyjście.

Nadzór polega na stworzeniu zestawu danych uczących, czyli par:

1. wejściowy obiekt uczący (np. wektor);
2. pożądana przez nadzorcę (nauczyciela) odpowiedź (np. jakaś konkretna wartość liczbowa).

Zadaniem systemu jest nauczenie się przewidywania prawidłowej odpowiedzi na zadane pobudzenie oraz generalizacja przypadków wyuczonych na przypadki, z którymi system jeszcze się nie zetknął. Do modelowania procesów technicznych zwykle wykorzystuje się uczenie nadzorowane.

Polega ona na dostarczeniu systemowi uczącemu się przykładów z których każdy składa się z dwóch elementów : informacji wejściowych do systemu i odpowiadających im informacji, które otrzymujemy na wyjściu. Innymi słowy mówiąc, mamy wektor x, którego elementy zwierają informacje dotyczące danego przykładu , które podajemy systemowi na wejście i wektor y który zawiera informacje pożądane na wyjściu. Zadanie systemu polega na nauczeniu się na podstawie informacji zawartych w wektorze x generowania zawartości wektora y. System powinien nauczyć się pewnej funkcji odwzorowującej wektor x na wektor y.

**Algorytmy bez nadzoru**

W tej klasie metod uczenia sztucznych sieci neuronowych (zwanej uczeniem bez nadzoru lub bez nauczyciela) sieć bez informacji zwrotnej korygującej jej działanie z zewnątrz (tj. oceny działania sieci) sama wypracowuje funkcje przetwarzania danych, np. uporządkowywania, wykrywania regularności w danych, ich klasyfikacji, kodowania itd.

Metody uczenia nienadzorowanego są ważnym narzędziem przetwarzania danych w sytuacjach gdy nie można zorganizować nadzorowanego procesu uczenia.

Uczenie nienadzorowane znane również jako uczenie bez nadzoru lub bez nauczyciela polega na dostarczenia do systemu zbioru przykładów opisanych jedynie za pomocą wektorów wejściowych x bez podania pożądanych odpowiedzi , czyli bez wektora y. Celem jest zazwyczaj przekształcenie dostarczanej do systemu uczącego się informacji wejściowej w pewne inne formy, lepiej dostosowane do dalszego przetwarzania. Zasada, zgodnie z którą jest dokonywane to przekształcenie, jest ustalona dla danej klasy zadań uczenia się bez nadzoru i wbudowana do struktury systemu. Można więc powiedzieć, że system uczący się bez nadzoru ma na stałe wbudowane źródło informacji trenującej. Typowym przykładem zadania uczenia się bez nadzoru jest zadanie wykrywania pewnych regularności w danych wejściowych i grupowanie ich na tej podstawie w pewne kategorie. Jest to tzw. grupowanie (klasteryzacja) oraz jej "bardziej symboliczna" forma, grupowanie pojęciowe (conceptual clustering). Algorytmy uczenia się bez nadzoru stanowią ważną grupę algorytmów uczenia sieci neuronowych.

**Algorytmy z częściowym nadzorem**

Algorytm z częściowym nadzorem jest używany gdy część danych uczących jest niezdefiniowanych.

Algorytmy te potrafią użyć dodatkowe niezdefiniowane dane by lepiej generalizować dane dla nowych próbek. Te algorytmy są najefektywniejsze gdy mają małą ilość zdefiniowanych danych i dużą ilość niezdefiniowanych.

: Semi-supervised learning methods use unlabeled data to either modify or reprioritize hypotheses obtained from labeled data alone. Although not all methods are probabilistic, it is easier to look at methods that represent hypotheses by p(y|x), and unlabeled data by p(x). Generative models have common parameters for the joint distribution p(x, y). It is easy to see that p(x) influences p(y|x). Mixture models with EM is in this category, and to some extent self-training. Many other methods are discriminative, including transductive SVM, Gaussian processes, information regularization, and graph-based methods. Original discriminative training cannot be used for semi-supervised learning, since p(y|x) is estimated ignoring p(x). To solve the problem, p(x) dependent terms are often brought into the objective function, which amounts to assuming p(y|x) and p(x) share parameters.

Biorąc pod uwagę sposób działania wyróżniamy wiele algorytmów miedzy innymi:

* algorytmy regresyjne
* drzewa decyzyjne
* algorytmy Bayesowe
* algorytmy grupowania
* sztuczne sieci neuronowe

Algorytmy regresyjne

n [statistical modeling](https://en.wikipedia.org/wiki/Statistical_model), **regression analysis** is a statistical process for estimating the relationships among variables. It includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between a [dependent variable](https://en.wikipedia.org/wiki/Dependent_variable) and one or more [independent variables](https://en.wikipedia.org/wiki/Independent_variable) (or 'predictors'). More specifically, regression analysis helps one understand how the typical value of the dependent variable (or 'criterion variable') changes when any one of the independent variables is varied, while the other independent variables are held fixed. Most commonly, regression analysis estimates the [conditional expectation](https://en.wikipedia.org/wiki/Conditional_expectation) of the dependent variable given the independent variables – that is, the [average value](https://en.wikipedia.org/wiki/Average_value) of the dependent variable when the independent variables are fixed. Less commonly, the focus is on a [quantile](https://en.wikipedia.org/wiki/Quantile), or other [location parameter](https://en.wikipedia.org/wiki/Location_parameter) of the conditional distribution of the dependent variable given the independent variables. In all cases, the estimation target is a [function](https://en.wikipedia.org/wiki/Function_(mathematics)) of the independent variables called the **regression function**. In regression analysis, it is also of interest to characterize the variation of the dependent variable around the regression function which can be described by a [probability distribution](https://en.wikipedia.org/wiki/Probability_distribution).

Regression analysis is widely used for [prediction](https://en.wikipedia.org/wiki/Prediction) and [forecasting](https://en.wikipedia.org/wiki/Forecasting), where its use has substantial overlap with the field of [machine learning](https://en.wikipedia.org/wiki/Machine_learning). Regression analysis is also used to understand which among the independent variables are related to the dependent variable, and to explore the forms of these relationships. In restricted circumstances, regression analysis can be used to infer[causal relationships](https://en.wikipedia.org/wiki/Causality) between the independent and dependent variables. However this can lead to illusions or false relationships, so caution is advisable;[[1]](https://en.wikipedia.org/wiki/Regression_analysis#cite_note-1) for example, [correlation does not imply causation](https://en.wikipedia.org/wiki/Correlation_does_not_imply_causation).

Many techniques for carrying out regression analysis have been developed. Familiar methods such as [linear regression](https://en.wikipedia.org/wiki/Linear_regression) and[ordinary least squares](https://en.wikipedia.org/wiki/Ordinary_least_squares) regression are [parametric](https://en.wikipedia.org/wiki/Parametric_statistics), in that the regression function is defined in terms of a finite number of unknown[parameters](https://en.wikipedia.org/wiki/Parameter) that are estimated from the [data](https://en.wikipedia.org/wiki/Data). [Nonparametric regression](https://en.wikipedia.org/wiki/Nonparametric_regression) refers to techniques that allow the regression function to lie in a specified set of [functions](https://en.wikipedia.org/wiki/Function_(mathematics)), which may be [infinite-dimensional](https://en.wikipedia.org/wiki/Dimension).

The performance of regression analysis methods in practice depends on the form of the [data generating process](https://en.wikipedia.org/wiki/Data_collection), and how it relates to the regression approach being used. Since the true form of the data-generating process is generally not known, regression analysis often depends to some extent on making assumptions about this process. These assumptions are sometimes testable if a sufficient quantity of data is available. Regression models for prediction are often useful even when the assumptions are moderately violated, although they may not perform optimally. However, in many applications, especially with small [effects](https://en.wikipedia.org/wiki/Effect_size) or questions of [causality](https://en.wikipedia.org/wiki/Causality) based on [observational data](https://en.wikipedia.org/wiki/Observational_study), regression methods can give misleading results.[[2]](https://en.wikipedia.org/wiki/Regression_analysis#cite_note-2)[[3]](https://en.wikipedia.org/wiki/Regression_analysis#cite_note-3)

Drzewa decyzyjne

**Decision Trees (DTs)** are a non-parametric supervised learning method used for [classification](http://scikit-learn.org/stable/modules/tree.html#tree-classification) and [regression](http://scikit-learn.org/stable/modules/tree.html#tree-regression). The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

For instance, in the example below, decision trees learn from data to approximate a sine curve with a set of if-then-else decision rules. The deeper the tree, the more complex the decision rules and the fitter the model.

Algorytmy Bayesowe

Algorytmy Bayesowe to takie które korzystają z teorii Bayesa dla problemów klasyfikacji czy regresji.

Dla dowolnej hipotezy h ∈ H i zbioru danych T ⊂ X zachodzi:

Pr(h|T) = Pr(T|h) Pr(h) Pr(T)

Algorytmy grupowania

Zadaniem algorytmów grupowania jest podział danego zbioru danych na mniejsze podzbiory. Elementy należą do danego podzbioru, gdy są bardziej podobne do innych elementow z tego podzbioru niż do obiektów z innych podzbiorów.

Podstawowym celem grupowania danych jest:

● wyszukiwanie wzorców,

● wskazywanie wspólnych cech poszczególnych obiektów w przestrzeniach wielowymiarowych,

● odnajdywanie podobnych obiektów,

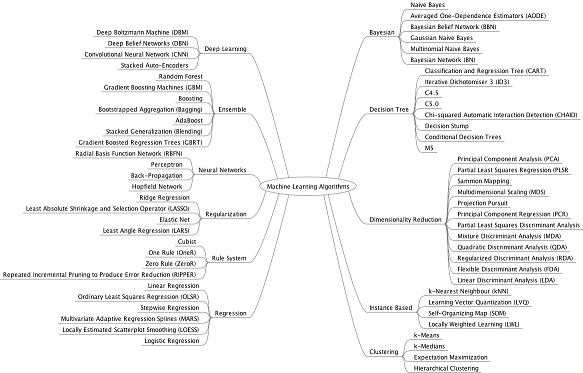
● łączenie obiektów w struktury hierarchiczne.

Sieci neuronowe

**S**ztuczne **S**ieci **N**euronowe (**SSN**) określa się najczęściej programowe lub sprzętowe symulatory modeli matematycznych, realizujące (pseudo-)równoległe przetwarzanie informacji, składające się z wielu wzajemnie połączonych funktorów zwanych poprzez analogię ze swoimi biologicznymi protoplastami - neuronami. Emulują one niektóre spośród zaobserwowanych właściwości biologicznych układów nerwowych oraz bazują na analogii adaptacyjnego uczenia biologicznego. Odnosząc tę definicję do architektury połączeń, równoległego przetwarzania i systemów neuromorficznych, sztuczne sieci neuronowe są swoistym systemem inspirowanym przez to, w jaki sposób gęsto połączone między sobą struktury mózgu obrabiają dane, w różny sposób docierające z otoczenia. Kluczowym elementem SSN jest zatem nowatorska struktura systemu przetwarzania informacji. System taki składa się z dużej liczby rozlegle połączonych ze sobą elementów przetwarzających, które - jak wspomniano wyżej - są analogiczne do biologicznych neuronów i powiązane ze sobą ważonymi połączeniami, które są znowu analogiczne do biologicznych synaps.

szczegółowy podział przedstawia poniższy rysunek

Rys. 1 podzial stylow uczenia się ze względu na sposób działania [http://machinelearningmastery.com/a-tour-of-machine-learning-algorithms Jason Brownlee 2013]]



# Część projektowa.

# Opis techniczny

Środowiska, gdzie system musi się dynamicznie dostosowywać do zmieniających się warunków (np. systemy sterowania)

# Przedmiot opracowania

Projekt samouczącego się systemu sterowania ogrzewaniem w domu jednorodzinnym. Zaprojektowano system który przewiduje godziny powrotu właściciela do domu oraz zapewnia odpowiednio wczesne uruchamianie ogrzewania w zależności od temperatury zewnętrznej oraz czasu nagrzewania domu. Celem systemu jest zapewnienie komfortu grzewczego użytkownikom domu.

* 1. Podstawa opracowania

Projekt opracowano w oparciu o:

* Temat pracy
* Środowisko programistyczne Visual Studio 2015
* Język programowania C#
* Bibliotekę Encog
* Literaturę , artykuły, strony internetowe ujęte w rozdziale X
  1. Charakterystyka obiektu

Jako obiekt badań przyjęto dom jednorodzinny drewniany zlokalizowany we Wrocławiu, na terenie miejskim, niezacienionym. Dom jest jednokondygnacyjny, ocieplony wełną mineralną do uzyskania współczynników przenikania ciepła na poziomie współczynników zgodnych z Rozporządzeniem Ministra Infrastruktury z dnia 12 kwietnia 2002 r. „W sprawie warunków technicznych jakim powinny odpowiadać budynki i ich usytuowanie” wraz z późniejszymi zmianami.

* 1. Założenia projektowe.

Założenia projektowe dla budynku:

* Powierzchnia domu: 100m2
* Powierzchnia ścian zewnętrznych: 98m2
* Powierzchnia przegród szklanych(okien):22m2
* Powierzchnia dachu: 100m2
* Powierzchnia podłogi na gruncie 100m2
* Ściany zewnętrzne drewniane, modrzewiowe +ocieplenie wełną mineralną
* Preferowana temperatura wewnętrzna 21stC
* Budynek jest ogrzewany piecem gazowym kondensacyjnym o mocy 15 KW
* Rodzaj ogrzewania grzejnikowe

Dla uproszczenia obliczeń przyjęto iż cały obrys budynku stanowi jedną powierzchnie o preferowanej temperaturze wewnętrznej 21stC. Dla powyższych założeń wyznacza straty ciepła dla budynku w zależności od temperatury zewnętrznej oraz czas nagrzewania budynku.

Założenia projektowe dla systemu sterowania ogrzewaniem:

* Sterowanie ogrzewaniem odbywa się za pomocą wbudowanego sterowania kotła grzewczego.
* Wbudowana automatyka kotła realizuje funkcję przeciwzamrożeniową
* Zaprojektowana aplikacja predykuje godziny powrotu właściciela do domu oraz oblicza czas nagrzewania i uruchamia kocioł o odpowiedniej godzinie.
* Do przewidywania godziny powrotu lokatorów do domu zastosowano algorytmy : Sztuczną sieć neuronową(Multi-Layer Perceptron) oraz Regresje liniową.
* Aktualną pogodę tj. temperaturę pobiera się z serwisu OpenWeatherService.
* Założono że sterowanie ogrzewaniem będzie odbywało się za pomocą biblioteki Modebus.
  1. Opis zastosowanych rozwiązań.
     1. Pobieranie pogody
     2. Sztuczna sieć neuronowa

Poprzez nazwę **S**ztuczne **S**ieci **N**euronowe (**SSN**) określa się najczęściej programowe lub sprzętowe symulatory modeli matematycznych, realizujące (pseudo-)równoległe przetwarzanie informacji, składające się z wielu wzajemnie połączonych funktorów zwanych poprzez analogię ze swoimi biologicznymi protoplastami - neuronami. Emulują one niektóre spośród zaobserwowanych właściwości biologicznych układów nerwowych oraz bazują na analogii adaptacyjnego uczenia biologicznego. Odnosząc tę definicję do architektury połączeń, równoległego przetwarzania i systemów neuromorficznych, sztuczne sieci neuronowe są swoistym systemem inspirowanym przez to, w jaki sposób gęsto połączone między sobą struktury mózgu obrabiają dane, w różny sposób docierające z otoczenia. Kluczowym elementem SSN jest zatem nowatorska struktura systemu przetwarzania informacji. System taki składa się z dużej liczby rozlegle połączonych ze sobą elementów przetwarzających, które - jak wspomniano wyżej - są analogiczne do biologicznych neuronów i powiązane ze sobą ważonymi połączeniami, które są znowu analogiczne do biologicznych synaps.

SSN są podzbiorem rozwiązań nazywanych AI (*Artifical Inteligence* - sztuczna inteligencja). Ich dynamiczny rozwój nastąpił dopiero od drugiej połowy lat 80, w wyniku możliwości jakie dała współczesna informatyka i elektronika.

Podstawową cechą różniącą SSN od programów realizujących algorytmiczne przetwarzanie informacji jest zdolność generalizacji czyli umiejętność uogólniania wiedzy dla nowych wzorców nieznanych wcześniej, czyli nie prezentowanych w trakcie nauki. Określa się to także jako zdolność SSN do aproksymacji wartości funkcji wielu zmiennych w przeciwieństwie do interpolacji możliwej do otrzymania przy przetwarzaniu algorytmicznym. Można to ująć jeszcze inaczej. Np. systemy ekspertowe z reguły wymagają zgromadzenia i bieżącego dostępu do całej wiedzy na temat zagadnień, o których będą rozstrzygały. SSN wymagają natomiast jednorazowego nauczenia, przy czym wykazują one tolerancję na nieciągłości, przypadkowe zaburzenia lub wręcz braki w zbiorze uczącym. Pozwala to na zastosowanie ich tam, gdzie nie da się rozwiązać danego problemu w żaden inny, efektywny sposób.

Uczenie w systemach biologicznych prawdopodobnie zmienia ustawienia w połączeniach synaptycznych, znajdujących się pomiędzy neuronami.

Uczenie SSN natomiast zmienia liczbowe wartości wag znajdujących się również pomiędzy neuronami. Uczenie zatem zachodzi poprzez bezpośrednią ekspozycję rzeczywistego zestawu danych, gdzie algorytm uczący modeluje wagi połączeń.Te właśnie wagi połączeń mają 'zapisane' dane niezbędne do rozwiązywania specyficznych problemów. Pomimo pojawienia się SSN w późnych latach '50, dopiero w połowie '80 stały się wystarczająco dojrzałe do zastosowania w poważnych aplikacjach. Dzisiaj SSN są stosowane do wzrastającej liczby problemów świata rzeczywistego o znacznym stopniu zawiłości. Wiele skutecznych predykcji uzyskanych za pomocą SSN opiera się na prezentacji ciągów uczących wygenerowanych z danych historycznych.

SSN oferują idealne rozwiązania dla dużego zakresu klasyfikowanych problemów (jak np.: mowa, rozpoznawanie znaków i sygnałów) równie dobrze jak predykcja i modelowanie systemów, gdzie procesy (fizyczne, ekonomiczne, ...) są niezrozumiałe bądź bardzo skomplikowane. SNN mogą być stosowane do nadzoru nad procesami przebiegającymi w czasie rzeczywistym, gdzie zmienne wejściowe są odczytami pomiarów używanymi do sterowania *on-line* , a sieć uczy się funkcji kontroli.

Siec neuronowa stara się przewidzieć przyszłe wartości opierając się na teraźniejszy oraz przeszłych wartościach. Do przewidywania wartości używamy temporal neural network. Temporal neural network jest zwykle feedforward network lub simple recurrent network.

Przewidująca sieć neuronowa korzysta z danych wejściowych do by zaakceptować informacje o aktualnych danych i używa ich do przewidzenia przyszłych danych. Korzystamy z dwóch okien:

-okna przeszłości

-okna przyszłości

Oba te okna muszą mieć określoną wielkość. Wielkość okna przeszłości określa jaka ilość danych jest potrzeba do wykonania predykcji. Wielkość okna przyszłości odpowiada ilości danych jakie mają zostać przewidzone. Dane przed przesłaniem do sieci neuronową powinny być znormalizowane zanim zostaną przekazane do sieci neuronowej.

- Wykorzystany algorytm.

Do zbudowania sieci neuronowej skorzystałem z opensourcowej biblioteki Encog która jest dostępna dla m.in. języków C# i Java.

By wykorzystać temporal neural network musimy zainicjalizować TemporalMLDataSet – jest to klasa w której będą przechowywane dane użyte do trenowania sieci. Wartości potrzebne do zainicjalizowania to wielkość okna przeszłości oraz przyszłości. Następnie musimy opisać jakie dane są podawane do sieci neuronowej. Ponieważ danę są wcześniej znormalizowane, jako typ danych podajemy że dane w sieci nie mają być przetworzone(typ danych RAW)

Następnie tworzymy zbiór punktów na których sieć będzie się uczyć.

Podstawową sprawą jest także stworzenie samej sieci.

Sieć neuronową dzielimy na warstwy. W naszym algorytmie skorzystamy z 3 warstw:

* Warstwy wejściowej o wielkości 10(Wielkość okna wejściowego)
* Warstwy ukrytej o wielkości 10
* Warstwy wyjściowej o wielkości 1(Wielkość okna wyjściowego)

Ilość neuronów w warstwie ukrytej została ustalona metodą prób I błędów gdyż nie jest znany algorytm pozwalający na obliczenie najlepszej ilości. Rozpocząłem od wartości minimalnej i zwiększałem ilość neuronów do zadowalających rezultatów

Dla każdej warstwy należy wybrać odpowiednią funkcje aktywacji (u nas tangens hiperboliczny?)

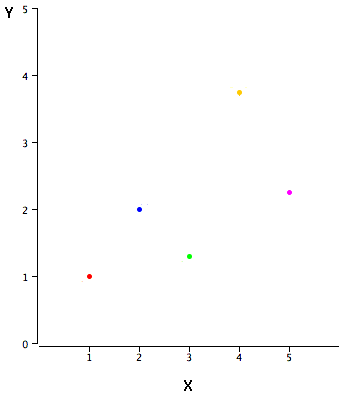
Gdy mamy już sieć i odpowiednie dane treningowe musimy wybrać odpowiedni algorytm trenujący sieć. W naszym przypadku korzystamy z jednego z najszybszych – Levenberg-Marquadt.

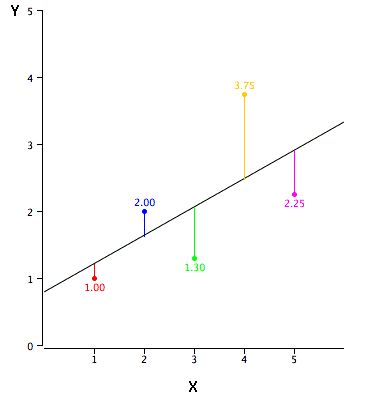
W tym momencie możemy zacząć trenować sieć. Sieć jest trenowana do momentu gdy błąd nie przekracza 0.001.

Gdy sieć skończy się uczyć możemy zacząc wykonanie przwidywania;

* + 1. Regresja liniowa

In simple linear regression, we predict scores on one variable from the scores on a second variable. The variable we are predicting is called the *criterion variable* and is referred to as Y. The variable we are basing our predictions on is called the *predictor variable* and is referred to as X. When there is only one predictor variable, the prediction method is called *simple regression*. In simple linear regression, the topic of this section, the predictions of Y when plotted as a function of X form a straight line.





<http://onlinestatbook.com/2/regression/intro.html>

Działanie regresji liniowej

* + 1. Obliczanie czasu nagrzania domu

# Opis badań

Aby zapewnić komfort cieplny użytkownikom budynku zaprojektowano system samouczący wyznaczający godziny powrotu lokatorów do domu. W tym celu zaprojektowano i porównano dwa algorytmy samouczące. Oba algorytmy potrzebują danych testowych do przewidywania wyniku.

Badania zostały przeprowadzone przez okres 2 tygodni. Podczas badań porównujemy wartości predykowane z obu algorytmów z rzeczywistymi wartościami powrotu do domu. Badania zostały przeprowadzone na przykładzie powrotów do domu 27 letniego Mężczyzny.

* 1. Dane testowe

Dane do przeprowadzenia badań pochodzą z okresu od 23.10.2015 do 26.05.2016. Algorytmy przewidujące zostały zasilone takimi samymi danymi by można wiarygodnie porównać wyniki badań. Dane są wyselekcjonowane dla danego dnia tygodnia.

* + 1. Sieć neuronowa

Sieć neuronowa potrzebuje co najmniej 40 danych by wykonać przewidywanie. Jest pewnie że na początku nie będziemy mieć możliwości podana wszystkich danych poprawnie, więc system doda potrzebne wartości losując z dostępnych wartości by była możliwość wykonania przewidywania. Dane dla sieci neuronowej zwiększamy co 10 gdyż algorytm jako okno wejściowe potrzebuje 10 danych.

* + 1. Liniowa regresja

Algorytm liniowej regresji zasilamy wszystkimi danymi jakie posiadamy.

# 4.Wyniki badań

W tabeli przedstawiono wyniki badań dla okresu 4 tygodni

Dla małej ilości danych sieć neuronowa źle się zachowuje, musi dolosować zbyt wiele wyników przez co wyniki są mało dokładne

# 5. Wnioski