Preamble: this is not a valid way to present theoretical part so I didn’t bother to format it. Also handwritten thing is not 1 to 1 copy – some reinterpretation is present.

1. preamble
   1. After third time rethinking the task this is how I understood: system gathers data for each ONE specific product and makes a decision of how many such items should be kept in stockpile. But then what’s stopping the system from just saying: “I want 50 of everything (1-50Eur range)”. So I also consider that there are user settings employed, that globally control price ranges, but also include things like stockpiling priority (keep more or less in stockpile than normally), is it a Christmas-ish type of item (works as a present (expecting more sales in December so stockpile more around December)), flag to omit from the auto stockpiling system, set some threshold and etc.
   2. Step one: get sales history, get search history, get service history, get current inventory, get user settings. Step two: make a decision based on gathered data.
   3. Make system that for each item could:
      1. Get sales history – how many and when were they sold.
      2. Get search history – how many times and when the item was searched for and view count.
      3. Get view history
      4. Get service history – see if item was returned or serviced previously.
      5. Provide UI so that system user could change settings for the item.
      6. Determine desired amount of item in question to be kept in stockpile based on data gathered above
      7. Take action based on existing quantity in stockpile of said item.
2. Now I’ll focus on task 5 out of before mentioned tasks, since determining stockpile amount is where the intelligent system could be used for and not in feature gathering or taking an action upon determining desired amount.
   1. Now the system has 5 features coming and some amount of setting set by user so there are few ways determining how many of that thing to keep:
      1. Basic approach: make a custom formula.
      2. MLP that in last layer has as many neurons as needed – 50 for 1-50Eur range and etc. By which I mean neuron that gives 1 indicates that that amount should be stockpiled
      3. MLP with non-linear transfer function on single last layer neuron. By which I mean neuron gives a range from 1 to max (0 to 1 really) stating the exact amount of units to keep in stockpile.
      4. NN plus fuzzy logic combination.
      5. SOM
   2. More in-depth look:
      1. Have all the features so just make a formula that accounts for all of them with varying weights – which effectively is a result of trained neural network… How things were done before neural networks. For simpler cases this is how I do things. Just expand a basic neural network formula and start fiddling with the parameters.
      2. From what I’ve experienced playing around in MATLAB, getting output to be 1 on only 1 output neuron of a lot is not easy and usually answer is not correct. So this method would need a bunch of data to train it and such. Gives a vector with a 1 at some index.
      3. Only difference from II is that this one gives a number of items to stockpile directly. And getting a vector form number or vice versa is not hard: **(enter script here)**

**If length(x)>1  
x=find(x==1)  
else  
t=zeros(1,max); t(x)=1; x=t;  
end**

* + 1. Looking for free scientific papers I did see some using combination of neural network and fuzzy logic. These were paid so don’t know how they implement it, but I figure they use NN for classifying result into some class and fuzzy logic states the certainty of that result belonging to said class
    2. SOM apparently can be used anywhere. Just give it all the features, tell how many items max is allowed and it’ll give you a number, sometimes a good one.

1. Ladies and gentlemen, engineers and time wasters I’m here to offer you the ultimate solution to your goods stockpiling problem! Don’t know how many items to stock up for next time period? Well then let this multi-layer perceptron based artificial neural network do the decision for you so that you can use your neural network for more important matters.
   1. Well I’ve wasted a bunch of time (2 hours), but found out how to use MATLAB to design basic feed-forward backdrop MLP, thus concluding that MLP with non-binary output from a single neuron is seemingly good enough for the task.
   2. Honestly though, first option, manual formula generation, wasn’t an option – too many variables. Fourth option is mostly guesswork how it works in the first place and, as I see it, it makes the result fuzzier when it actually needs to give precise-ish integer number. Second option was real contender for a time, but speed is not an issue in this case so transfer function may as well be more complex (not a step/linear function).  
      So non-linear transfer function MLP is precise, relatively simple and does not require much maintenance to make it work (few time periods to train up weights and biases).
2. How I image this system it would start working correctly after quite a few time periods, but to speed it up inputs VI and VII for a new item could be copied over from other similar item with already trained system. This is how I imagine the NN implementation per item (does not specify exact means of data acquisition or end decision implementation):
   1. Inputs:
      1. Latest time period sales – how many were sold.
      2. Latest time period search appearances – how many times search engine pulled this item for its results.
      3. Latest time period view count – how many times its details were viewed.
      4. Latest time period service data – how many were returned or serviced.
      5. Leftover stockpile – how many item left from previous time period
      6. Previous time periods data arrays – stored data of aforementioned inputs plus how many items ideally should’ve stockpiled for that period.
      7. Neural network parameters – weights and biases.
      8. Applicable user settings – which price/stockpile size range it falls into; is it Christmasy type thing and its December coming up; should it be over-stockpiled, under-stockpiled or no additional weighing and etc.

Outputs:

* + 1. Number of items there should be stockpiled.
  1. Action flow:
     1. If inputs VI and VII are available, train/adjust NN parameters with the old data.
     2. Simulate output with the updated NN.
  2. Well I used nntool for testing, but basic 2 layer MLP functions are applicable here. First layer output is , where is input value, n is input number, is hidden layer perceptron number, are weights, is bias. ϕ is the activation/transfer function. I tested with , but since negative numbers are not needed and to avoid shifting it all up or some other workaround/fix/improvement can be used. Formula for second layer outputs calculation is , where r is second/output layer neuron number, could be omitted since this system would use only one output perceptron, but this is general expression.  
     As for parameter updates: new last layer weights are found using , where is rate of change and is error obtained by using , where T is desired output or in this case amount of goods shop should’ve stockpiled for that period of time. Bias for second layer is attained using . And lastly, hidden layer weights update: ; in case of bias there’s just no .  
     final decision making I leave up to the programmer and client to specify but in general quantity requested to stockpile would still undergo this formula: , where is amount requested, amount given by NN and is surplus currently in stockpile.

Notes: first day I was depressed thinking I don’t understand it at all and couldn’t find the drive to start it. Second day morning started deciphering and thinking fully about it. Two hours later finally think understood the task, but also hit “whatever” stage and as such this might not be most sciency thing and not up to par with works that I did not gave up on but after 8 hours of work, don’t think it’s that bad. Also I’m noticing I have memory issues – honestly already don’t remember 90-98% of this semester; so in places I guess it might seem that I’m just throwing terms around, but I know what I mean just don’t know how to explain or misuse a term. As for search in internet for similar things: upon cursory search all relevant articles are in ScienceDirect and I’m not paying for them.