Hydropower Optimization Using Deep Learning

Bernt Viggo Matheussen[1](✉), Ole-Christoffer Granmo[2], and Jivitesh Sharma[2]

[1] Agder Energi AS, University of Agder, Grimstad, Norway
bernt.viggo.matheussen@ae.no
[2] Centre for Artificial Intelligence Research, University of Agder, Grimstad, Norway
{ole.granmo,jivitesh.sharma}@uia.no
https://www.ae.no, https://www.uia.no, https://cair.uia.no

Abstract. This paper demonstrates how deep learning can be used to find optimal reservoir operating policies in hydropower river systems. The method that we propose is based on the implicit stochastic optimization (ISO) framework, using direct policy search methods combined with deep neural networks (DNN). The findings from a real-world two-reservoir hydropower system in southern Norway suggest that DNNs can learn how to map input (price, inflow, starting reservoir levels) to the optimal production pattern directly. Due to the speed of evaluating the DNN, this approach is from an operational standpoint computationally inexpensive and may potentially address the long-standing problem of high dimensionality in hydropower optimization. Further on, our method may be used as an input for decision-theoretic planning, suggesting the policy that will give the highest expected profit. The approach also permits for a broader use of pre-trained neural networks in historical reanalysis of production patterns and studies of climate change effects.

Keywords: Scheduling · Markov · Hydrology · Reservoir · Energy · Water

1 Introduction

Operations of a multi-reservoir hydropower river system are challenging in many aspects. Depending on the system configuration, production planners must consider factors such as price developments, flooding risks, environmental constraints, and hydrological information, before they can decide upon the schedule of the hydropower production. Further on, forecasts of future inflow and price are highly uncertain. Additionally, the dynamics of a power system is non-linear. These properties make it difficult to optimize reservoir management. Indeed, the core of the problem involves a complex high-dimensional non-convex state space search. To cope with this complexity, researchers have for many decades developed optimization algorithms that can be used for decision support in daily hydropower operations [1]. Among the many techniques available, various linear

F. Wotawa et al. (Eds.): IEA/AIE 2019, LNAI 11606, pp. 110–122, 2019.
https://doi.org/10.1007/978-3-030-22999-3_11

and dynamic programming based methods seem to have gained a momentum compared to other methods. Some examples of these methods are linear programming (LP) [2,3], non-linear programming (NLP) [4,5] and dynamic programming (DP) [6,7].

Unfortunately, traditional methods have severe weaknesses. Most notably are the well know problems referred to as the "curse of dimensionality and modeling" [13]. Linear programming schemes, for instance, face difficulties with uncertain information, requiring extensive Monte Carlo simulation. Furthermore, it is difficult to capture non-linear relationships, e.g., the hydraulic head may not be represented correctly. Further, computational complexity multiplies with both the number of system components and the extent of the time horizon. Dynamic programming, on the other hand, handles input uncertainty. They scale linearly in the time dimension and capture non-linear and uncertain relationships. Such schemes generally require that models and system dynamics are fully known, at least at a probabilistic level. Further, computational complexity grows exponentially with the number of system components, making this class of schemes incapable of modeling larger systems in full complexity and detail. Temporal resolutions are often coarse (weekly, monthly), which results in an underestimation of flooding in smaller and steep catchments, resulting in sub-optimal solutions. Finally, in an operational setting, computation time is crucial for decision making, since the allocation of production units are synchronized with the physical markets. Due to their high computational demand, the classical techniques are slow to run, which is unpractical in time-constrained operations.

Several authors have applied meta-heuristic methods to mitigate some of these challenges. A recent paper [8] applies the so-called Firefly algorithm (FA) to the hydropower optimization problem. The algorithm is motivated by the grouping behavior of fireflies and has strong similarities to the particle swarm method [13]. The firefly technique is applied to the reservoir operation problem, and the authors claim that the Firefly algorithm is superior compared to genetic algorithms (GA). Another modern meta-heuristic method is the invasive weed optimization (IWO) algorithm [15]. IWO is a novel evolutionary algorithm inspired by colonizing weeds. Other methods include search algorithms such as gradient ascent/descent and simulated annealing. For more on this see [1,15,16] and citations therein.

Machine learning (ML) has further been used in water resources management. Labadie [1] refers to work starting in the 1970s shaping ideas on how to incorporate various ML concepts into the reservoir management problem. After that, a wide range of methodologies have appeared in the literature trying to improve either deterministic or stochastic methods. For example, DeRigo tried to use neural networks to estimate the Bellman function in stochastic dynamic programming [9]. Lee [10] demonstrated how the Q-Learning method in reinforcement learning was used on a two-reservoir river system. They claim that the method outperformed more classical techniques. Such an approach was further investigated and modified by [11] who used a tree-based reinforcement learning algorithm trying to identify optimal water reservoir operation for the Lake Como river system in Italy. The results showed improved performance compared to traditional (stochastic) DP methods. More recently Dariane used neural networks and reinforcement learning in combination with a meta-heuristic method (particle swarm) to optimize a river-system in Iran [13]. Lately, Sangiorgo used a NN-based ISO technique to optimize the Nile multi-reservoir system [14]. The results demonstrated among other things that NNs can reduce computational demand, facilitating real-time operation.

From the literature cited, we can see that different ML techniques have been applied to reservoir management problems around the world. Nevertheless, the application of these techniques to river systems located in the Nordic region is absent in the literature. The reason for this is not apparent, but in the Nordics, all power stations are connected to a competitive electricity market; thus both price and inflow must be accounted for in the optimization procedure. To our knowledge, the inclusion of price and inflow in an ML-based optimization algorithm has not been investigated before, for the Nordic region. It should be noted that the industry has used decision support tools based on the well-known LP and DP methods for more than a decade [3].

In this research we propose a modified ISO framework that has similarities to the work of [1,12–14]. Our method combines the use of historical data, synthetic time series, meta-heuristic optimization based on multi-start gradient ascent, and neural networks (NN), in an intricate interplay, explained in the next section. The ISO framework was chosen for several reasons. First of all, the hydraulics of a hydropower river system is strongly non-linear, so it was desirable with a method that could handle non-linearities in the optimization step. It is known that LP can find global optima for linear problems, but if the problem is not linear in the first place, then it may be questioned whether LP methods find "real-life" global optima. One of the advantages NNs provide in this setting is their low computational demand in operation after training has been completed. With proper tuning, they can run sufficiently fast on almost any device. Lastly, due to inflow and price forecasts being highly uncertain, our method must be able to deal with stochastic input. Our unique combination of methods supports the inclusion of price and inflow in the optimization and the use of neural networks in a theoretical decision planning concept. The method handles uncertainty (ensemble of price and inflow) and is based on a continuous model (as opposed to a discrete representation of states and actions). Such an approach has to our knowledge not been tested on a multi-reservoir system previously.

The rest of this paper is organized as follows. We first describe, in detail, the mathematical methods we propose using to resolve the hydropower optimization problem. Then we present the study area and available data used for evaluating the method. Finally, we discuss our results and provide conclusions and pointers for further research.

2 Method

The objective of this research is to develop a stochastic hydropower optimization algorithm that can maximize expected profit given an ensemble of forecasts of future reservoir inflow and market price. All relevant constraints and initial conditions must be taken into account. To do this we apply a modified version of the implicit stochastic optimization (ISO) method described in [1,12–14]. Figure 1 illustrates the components of our proposed architecture, inspired by the ISO framework.

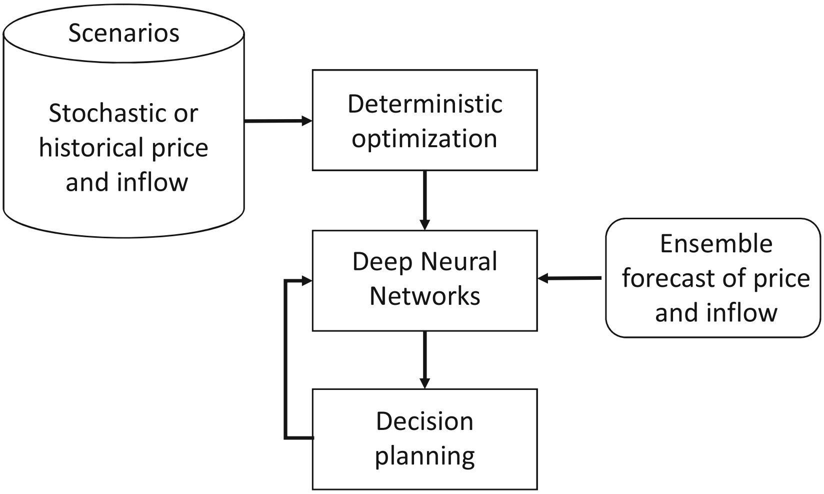
Scenarios
Stochastic or historical price and inflow
Deterministic optimization
Deep Neural Networks
Ensemble forecast of price and inflow
Decision planning

Fig. 1. Schematics of our implicit stochastic optimization (ISO) architecture, drawing upon previous work on ISO [1,12–14].

The first step in the method is to build Markov chains (MC) from the historical data of price and inflow. After that, we can sample from the Markov chains to generate an unlimited amount of scenarios with a given temporal resolution (e.g., hourly or daily) and planning horizon (e.g., 40 days). For each of these scenarios of price and inflow, we use a multi-start gradient ascent method to find the optimal operating policies. The reason for choosing this method was that it is a commonly used method in ML, it can find close to near-optimal solutions for many different problems, and it is easy to implement. It should be noted that any other deterministic optimization method can be used for this purpose (e.g., meta-heuristic methods). After a large number of scenarios have been generated and optimal operating policies for these scenarios have been found, we can use this input to train deep neural networks. In this work, we use dense multi-layer perceptron networks to make a mapping between the input scenario and the optimal operating policies. This mapping can then be used in a two-step procedure we refer to as "Decision planning". In all brevity, the first step of the decision planning is to use the DNNs to get a first estimate of the optimal policy for each of the ensemble of scenarios. Second, for each of the decisions found, we simulate the effect of this decision in one time step. Then we find the expected profit for the rest of the time horizon using another DNN. This allows us to effectively identify robust near-optimal strategies, as detailed in the following.

2.1 Time Series Generation with Markov Chains

For our time series generation module, we assume that there exist historical data of price and inflow for the hydropower system under consideration. The data are reported with a specified temporal resolution (e.g., daily time steps), but are continuous-valued. Based on the spread (min, max) and from visual inspection, the discrete time continuous-valued data may be turned into a discrete-valued sequence. From this we can compute the one-step state-transition probability matrix, defining a Markov Chain (MC) [16]. We use this matrix to sample an arbitrary amount of time sequences for training the DNN. To make the training data more diverse, we further added uniform noise to the discrete values, rendering the sampled sequences continuous-valued.

We define a scenario to be a discrete-time-continuous-valued sequence of a specified time horizon (T, e.g., 40 days) with price and inflow data. Besides this, a scenario also consists of initial reservoir levels as well as the expected power price at the end of the planning horizon (named rest price throughout the rest of this paper). An ensemble of such scenarios is illustrated in Fig. 3.

2.2 Deterministic Optimization of Operation Policies

For each of the scenarios sampled from the MCs, the optimal deterministic operating policy (i.e., water release decisions) must be identified. In this work, we decided to use a multi-start, one-at-a-time empirical gradient ascent method [16].

$$\nabla F = \left(\frac{\partial F}{\partial d_{1,0}}, \frac{\partial F}{\partial d_{1,1}}, \frac{\partial F}{\partial d_{2,0}}, \frac{\partial F}{\partial d_{2,1}}, \cdots \frac{\partial F}{\partial d_{T,0}}, \frac{\partial F}{\partial d_{T,1}} \right) \tag{1}$$

The method involves finding the empirical gradient illustrated in Eq. 1 changing one parameter (hatch release or production level) for one time step at a time. After that, we check the change in profit and then resetting it back to original before checking next time step. After all time steps have been tested we find the time step with the highest change in total profit. After this, we repeat it all over. This procedure is then started with several different initial values of the policies (hatch release and production level).

Equations 2 to 7 defines the deterministic optimization problem that must be resolved for each of the scenarios (s). For each hydraulic node in the system (n), profit (π) is calculated as the income ($e_t * w_t$), then subtracting the operational costs (c). We also model the effect of an operation policy that violates operational constraints such as minimum reservoir levels, or start-stop costs of machinery. In practice, this would also include minimum flow requirements and other environmental constraints, but this is not included in this work. The value of the remaining water (I) in the reservoirs, at the end of the planning horizon, is estimated assuming a rest price (e_{T+1}), and an average reservoir height. The value is added to the total profit for each of the hydraulic nodes (power station or reservoir) in the system. This optimization (maximization) is carried out for each scenario (s) in the ensemble.

$$Max(F), \qquad F = \sum_{n=1}^{N} \pi_{n,s}(d) \tag{2}$$

$$\pi_{n,s}(d) = \begin{cases} \sum_{t=1}^{T} \big(e_t\, w_t - c_t\big) + I(x_{T+1}, w_{T+1}, e_{T+1}) & \text{if } powerstation \\ \sum_{t=1}^{T} \big(-c_t\big) + I(x_{T+1}, w_{T+1}, e_{T+1}) & \text{if } reservoir \end{cases} \tag{3}$$

Subject to:

$$w_t = \kappa(\eta\rho g(h(x_t) - h_{ps} - \gamma d_{t,n}^2)d_{t,n}) \tag{4}$$

$$d_{min,n} \leq d_t \leq d_{max,n} \tag{5}$$

$$x_{n,min} \leq x_{n,t} \leq x_{n,max} \tag{6}$$

$$x_{n,t} = x_{n,t-1} - \delta d_{t,n} + \delta \sum_{j=1}^{N_u} (o_{j,t} + q_{j,t} + d_{j,(t-\tau)}) \tag{7}$$

where:

e_t = Price for energy [Euro/MWh]
e_{T+1} = Rest price: expected price after the planning horizon [Euro/MWh]
w_t = Power production [MWh]
c_t = Costs for violating reservoir levels, or start-stop [Euro]
η = Turbine and generator efficiency [fraction, unit free]
ρ = Density of water [1000 kg/m^2]
g = Gravity [9.81 m/s^2]
h = Water elevation height at reservoir or power station [m]
γ = Friction loss coefficient [s^2/m^5]
$d_{t,n}$ = Water release decision from powerplant or reservoir [m^3/s]
$x_{n,t}$ = Filling in reservoir n, [m^3]
I = Expected profit of residual water [Euro]
q = Natural inflow to node [m^3/s]
o = Overflow from upstream node (reservoir) [m^3/s]
T = Number of time stages
N = Number of nodes, (N_u number of upstream nodes)
κ = Daily time step conversion factor (24*3600)/(1000000*3600)
δ = Time step length (24*3600) [s]

2.3 Decision Planning

In this work, we assume that production planners are using an ensemble of price and inflow scenarios to identify the optimal water release decisions for a given hydropower system. Despite the availability of an ensemble of scenarios, enhanced with optimal decisions, making the correct decision is still difficult. The reason is that we do not know in advance which scenario actually will play out. Instead, we get a possibly wide range of plausible decisions, dependent on the input scenario. To resolve the problem of singling out the best decision to make, given an uncertain future, we apply a two-step procedure. Firstly, for all the scenarios in the ensemble ($s \in S$), we use the previously trained neural networks to find the associated optimal release decisions ($d \in D$). This provides a distribution over plausible optimal decisions conditioned on the ensemble data. Only the release decisions in the first time step are considered, even the deterministic optimization algorithm have considered the whole planning horizon when generating training data. Secondly, for each of the release decisions, we realize the first time step, update the reservoir content (Eq. 7), and then expected (mean) profit for all scenarios is calculated with another trained neural network. In general, this means finding the release decisions for the first time step that gives the highest expected stochastic profit after the first time step has been realized. This is shown in Eq. 8. Since we are using neural networks that have already been trained in previous steps, the computational requirements are relatively low.

$$\underset{d_{t_1} \in \mathcal{D}}{\operatorname{argmax}}\, E[\pi \mid d_{t_1}] = \{d_{t_1} \mid d_{t_1} \in D \wedge \forall d^* \in D : E[\pi \mid d^*] \leq E[\pi \mid d_{t_1}]\} \tag{8}$$

To resolve Eq. 8 efficiently, the NNs (Fig. 1) must be trained to provide a mapping between inflow-price and the various release decisions. Also needed is a network that can provide the maximum profit for a given scenario. The reason for this is that Eq. 8 requires that each scenario's profit is estimated. From an operational point of view, the use of neural networks replaces the need for deterministic optimization using time-consuming heuristic methods. In this work, we study a simple "two-node" system, which requires three neural networks to be trained (hatch release, production, profit) – more on this in the results section.

3 Study Area and Available Data

In this research the hydropower system of Kvinesdal, located in southern-Norway, is used as a case study. A location map, and schematics of the system is shown in Fig. 2. The system consists of two reservoirs and one hydropower station. The uppermost reservoir, named Tjeldaasvatn, is laying at an average elevation of 312 m above sea level (m.a.s.l). The reservoir is connected to the downstream reservoir Stampetjonn through an open channel. The water release out of Tjeldaasvatn is controlled by a hatch that can release up to 1 m^3/s of

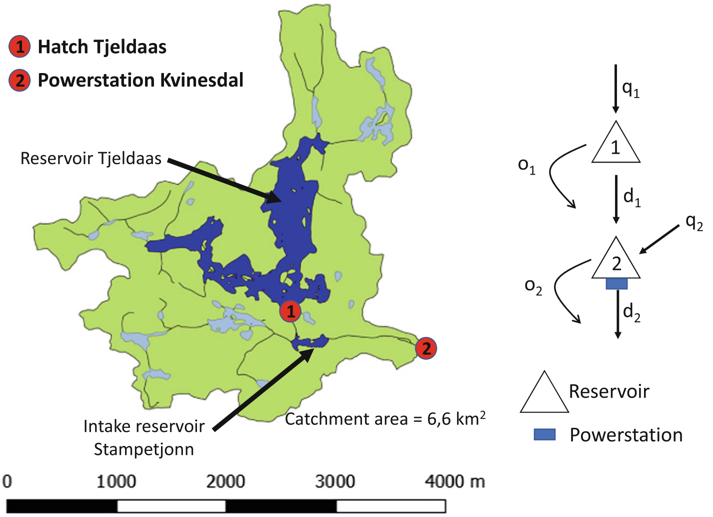

Fig. 2. Schematic of the Kvinesdal hydropower river system (southern-Norway).

water. The lowermost reservoir Stampetjonn serves as an intake reservoir to the Kvinesdal power plant. It is located at an average elevation of 302 m.a.s.l. Both the upper and lower reservoirs are constructed with overflow safety spillways that transports water into the old river bed during high water levels (above upper regulated water heights). During operations, water from the intake reservoir is transported through a $\sim$ 1 km circular pipe down to the power station at 38 m.a.s.l. Kvinesdal power plant has an installed maximum power capacity of 1.4 MW at a water usage of around 0.69 m^3/s. In Norway, this is considered to be a small station. During the period from 2007 until 2018 the power station produced energy for more than 90% of the time. This indicates relatively high water availability for the system. The powerplant is owned and operated by Agder Energi AS - a Norwegian provider of renewable energy. The plant is connected to the Nordic electricity grid and is part of the Nordic physical electricity market.

In this research, time series of price and inflow to the Kvinesdal system was provided from the historical archive of Agder Energi. The historical records covered the time period from January 1, 1996, until August 31, 2017, and had a temporal resolution of 24 h. The Pearson correlation (PC) between price and inflow was calculated to be −0.042, for the daily data over the time period 2007–2018. Data treated with a moving average filter (180 days) had a PC of 0.041. Based on this, it was assumed that there is neither dependence nor correlation between the market price and the inflow data. Thus, they can be treated separately for the rest of this work. It should be noted that the inflow data used in this work represents the inflow to the whole river system (catchment area 6.6 km^2). In the hydraulic calculations of the river system, local inflow to the system must be provided for both the upper and the lower reservoirs. This was resolved by splitting the inflow data into two time series scaled after the contributing area to the reservoirs. An effect of this is that both the upper and lower reservoirs receive local natural inflow at the exact time. Due to this research being a proof-of-concept, it was chosen to neglect this effect.

4 Results

The methodology used in this research involved training and testing of three neural networks (NN). These were all designed to make a mapping between the input (price and inflow of scenario), and output (hatch release, production level, and total profit) for a scenario. It was decided to use classical multi-perceptron fully connected (dense) feedforward networks. The input to the networks was chosen to be forty (thirty-nine for profit) days with inflow, price and the rank of the price. Besides the time series, rest price and initial fillings of the two reservoirs were also used as input to the NNs. A total of 116400 input scenarios with price, inflow, and so on, were prepared by sampling from the Markov chains fitted to the historical time-series data. All the scenarios were optimized with the multi-start gradient ascent method and the resulting policies (water release from the hatch, production level, profit) were used as output values (supervisory signal) to the neural networks. For hatch release and production level, data only for the first day of the planning horizon was used as the supervisory signal.

All the input data were normalized to have values between zero and one. Each neural network used five hidden layers in addition to the input and output layers. Hyper-parameters, width and depth and choice of activation functions were found by trial and error. A combination of hyperbolic tangents (tanh), rectified linear units (relu) and sigmoid activation functions showed the highest score on the objective criteria. It was decided to use 90 % of the available data as training for the neural networks and the remaining part for testing. The networks were trained using the RMSProb gradient algorithm [17] updated with the back-propagation method, optimizing for Mean Square Error (MSE) between the predictions and the supervisory data.

Table 1. Values of the objective criteria obtained during training and testing of the neural networks.

	Training				Testing			
Obj.crit	MSE	P.corr	MAE	Bias	MSE	P.corr	MAE	Bias
Prod	0.014	0.937	0.066	-0.001	0.025	0.889	0.089	-0.001
Hatch	0.012	0.954	0.057	-0.004	0.036	0.863	0.096	-0.001
Profit	0.000	0.999	0.005	0.002	0.000	0.999	0.005	0.002

Table 1 shows the results from the training and testing of the three deep neural networks (Prod., Hatch, Profit). In addition to MSE, three other objective criteria were used to quantify the performance of the NNs, i.e., Pearson correlation coefficient (P.corr), mean absolute error (MAE) and bias (difference in average). In general, it can be seen that the MSE is below 0.014 for the training period and below 0.036 for all the three networks in the testing period. The Pearson correlations are all around 0.94 and above. The MAE is around 0.06, and lower and bias is under 0.07. It can also be seen that the objective criteria are in general lower for the testing period. This is as expected since we test with data that has not been seen by the training algorithm. The results show that it is possible to make a mapping between inflow and price information and optimum production patterns for a hydropower system.

Figure 3, (a) and (b), illustrates an ensemble of price and inflow forecasts (scenarios) representative for the Kvinesdal hydropower system. The ensemble forecast has a planning horizon of forty days and was made available through the internal forecasting systems used by Agder Energi. The data were generated by the use of meteorological, hydrological and physically based power system models (price models). Due to time constraints and page limitations in this paper, it was chosen to neglect further details on how the forecasts were made. It was decided to use these data as external test data, assuming that they represent an actual stochastic forecast.

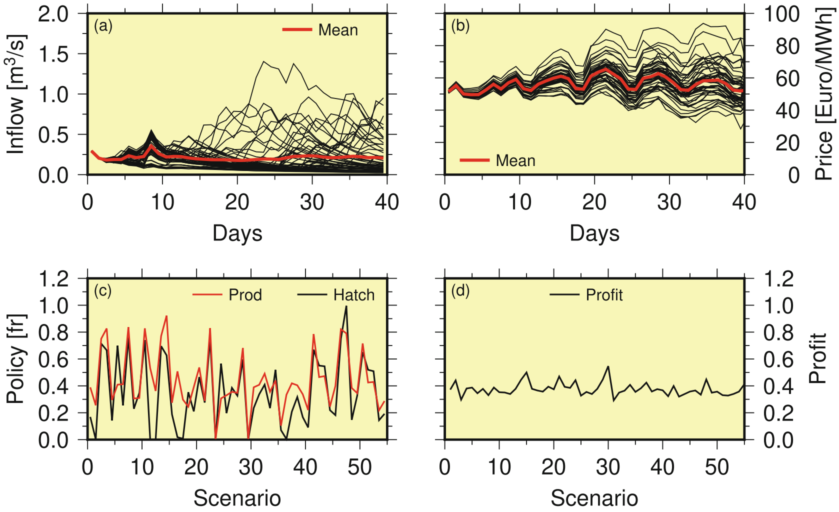

Fig. 3. Example of a real world ensemble forecast of inflow (a) and powerprice (b). Results from the decision planning method is shown in (c) and (d).

The ensemble forecast shown in Fig. 3, (a) and (b), were used as input to the decision theoretic planning approach described in earlier sections, and mathematically shown in Eq. 8. The method first calculates optimum hatch and production policies for each scenario using the Prod. and Hatch NNs. The results from this calculation is shown in Fig. 3(c). Secondly each of the optimum policies for the first day is then realized and new values of reservoirs levels are calculated after the first day. After that, the profit for all scenarios are computed for each of the policies, and the expectation (average profit) is calculated. The scenario with the highest profit corresponds to the policy that should be chosen as decision. In Fig. 3(d), the profit expressed as a fraction between zero and one is shown. It can be seen that it is for scenario 30 that we find the highest expected profit. In Fig. 3(c), we can see that this scenario has zero policy for both hatch and production. So the decision for the input ensemble shown in Fig. 3(a) and Fig. 3(b), would be 0.0 release of water from the upper to the lower reservoir. At the same time we should not produce since the policy for Prod is zero.

5 Discussion and Conclusions

This paper demonstrates how deep learning can be used together with relatively simple search algorithms to find optimal reservoir operating policies in hydropower river systems. The method is based on the ISO framework which uses ensemble input to make a stochastic optimization for a given system. The findings suggest that deep NNs can learn how to map input (price, inflow, starting reservoir levels) to the optimal production pattern directly. This approach is from an operational standpoint computationally in-expensive and may be utilized in several ways in the future. First, the NNs may be used to provide starting policies for metaheuristic optimization of longer time horizons or scenarios with higher temporal resolutions. This may potentially disseminate the problem of "curse of dimensionality". Although, as pointed out by Dariane [13], if the neural networks are trained indirectly through the use of metaheuristic algorithms, they may be slow to run for complex systems with long time horizons. Secondly, historical re-analysis of optimum production patterns is hard to do with classical methods, due to extreme computational demands, but this is not an issue when using NNs. Such abilities can be useful when studying the effects of climate change on hydropower operations since long time series must be used in climate studies. Further, since the proposed method has a distinction between the hydraulic simulator and the neural networks, there are no limitations in what type of constraints that can be included in the model. Such constraints may be ramping restrictions on how fast water release are allowed to change, it could also be minimum flow requirements or even identification of optimal water release for salmon habitat.

One weakness in this work is that the proposed optimization algorithm has only been tested for one example data set. In the future, the method must be tested for a larger number of cases and be benchmarked against other stochastic optimization algorithms. It is a paradox that the real global optimum for many situations may never be found although we run multiple algorithms on powerful computers. The methods we apply to the reservoir problem may only be tested and benchmarked against each other. Further on, the proposed method should also be tested on more complex hydropower system, with several more reservoirs and power stations, to see if it scales. A coarser temporal resolution should also be used in the analysis.

Acknowledgements. This research has been supported by Agder Energi, the Norwegian Research Council (ENERGIX program), and the University of Agder. We also thank Jarand Roeynstrand at Agder Energi for input on an early version of this article.

References

1. Labadie, J.W.: Optimal operation of multi-reservoir systems: state-of-the-art review. J. Water Resour. Plann. Manag., 93 (2004). https://doi.org/10.1061/(ASCE)0733-9496(2004)130:2(93)
2. Yoo, J.-H.: Maximization of hydropower generation through the application of a linear programming model. J. Hydrol. **376**, 182–187. https://doi.org/10.1016/j.jhydrol.2009.07.026
3. Belsnes, M.M., Wolfgang, O., Follestad, T., Aasgård, E.K.: Applying successive linear programming for stochastic short-term hydropower optimization. Electr. Power Syst. Res. **130**, 167–180 (2016)
4. Moosavian, S.A.A., Ghaffari, A., Salimi, A.: Sequential quadratic programming and analytic hierarchy process for nonlinear multiobjective optimization of a hydropower network. In: Optimal Control Applications and Methods. https://doi.org/10.1002/oca.909
5. Jothiprakash, V., Arunkumar, R.: Multi-reservoir optimization for hydropower production using NLP technique. KSCE J. Civ. Eng. **18**(1), 344–354 (2014)
6. Allen, B., Bridgeman, R.G.: Dynamic programming in hydropower scheduling. J. Water Resour. Plann. Manag., ASCE. https://doi.org/10.1061/(ASCE)0733-9496(1986)112:3(339)
7. Zhao, T., Zhao, J., Yang, D.: Improved dynamic programming for hydropower reservoir operation. J. Water Resour. Plann. Manag. **140**(3), 365–374 (2012)
8. Garousi-Nejad, I., Bozorg-Haddad, O., Loáiciga, H.: Modified firefly algorithm for solving multi-reservoir operation in continuous and discrete domains. J. Water Resour. Plann. Manag. 142(9) (2016). https://doi.org/10.1061/(ASCE)WR.1943-5452.0000644
9. DeRigo, D., Rizzoli, A.E., Soncini-Sessa, R., Weber, E., Zenesi, P.: Neuro-dynamic programming for the efficient management of reservoir networks. In: Proceedings of MODSIM 2001 International Congress on Modelling and Simulation, Vol. 4, pp. 1949–1954. Modelling and Simulation Society of Australia and New Zealand, December 2001. ISBN 0-867405252
10. Lee, J.H., Labadie, J.W.: Stochastic optimization of multi-reservoir systems via reinforcement learning. Water Resour. Res. **43**, W11408 (2007). https://doi.org/10.1029/2006WR005627
11. Castelletti, A., Galelli, S., Restelli, M., Soncini-Sessa, R.: Tree-based reinforcement learning for optimal water reservoir operation. Water Resour. Res. **46**, W09507 (2010). https://doi.org/10.1029/2009WR008898
12. Loucks, D.P.: Implicit stochastic optimization and simulation. In: Marco, J.B., Harboe, R., Salas, J.D. (eds.) Stochastic Hydrology and its Use in Water Resources Systems Simulation and Optimization. NATO ASI Series (Series E: Applied Sciences), vol. 237. Springer, Dordrecht (1993). https://doi.org/10.1007/978-94-011-1697-8_19

13. Dariane, A.B., Moradi, A.M.: Comparative analysis of evolving artificial neural network and reinforcement learning in stochastic optimization of multi-reservoir systems. Hydrol. Sci. J. **61**(6), 1141–1156 (2016). https://doi.org/10.1080/02626667.2014.986485
14. Sangiorgio, M., Guariso, G.: NN-based implicit stochastic optimization of multi-reservoir systems management. Water **10**, 303 (2018)
15. Azizipour, M., Ghalenoei, V., Afshar, M.H., Solis, S.S.: Optimal operation of hydropower reservoir systems using weed optimization algorithm. Water Resour. Manag. **30**, 3995–4009 (2016). https://doi.org/10.1007/s11269-016-1407-6
16. Russel, S., Norvig, P.: Artificial Intelligence: A modern Approach, 3rd edn., pp. 131 or 589. Prentice Hall. ISBN 13 978-0-13-604259-4
17. Keras documentation. https://keras.io/optimizers/